AD A057506

AFGL-TR-78-0081

LEVEL II

(10)

FASCODE - FAST ATMOSPHERIC SIGNATURE CODE
(SPECTRAL TRANSMITTANCE AND RADIANCE)

H.J.P. Smith
D.J. Dube
M.E. Gardner
S.A. Clough
F.X. Kneizys
L.S. Rothman

Visidyne, Inc.
19 Third Avenue
Burlington, Massachusetts 01803

Scientific Report No. 2

16 January 1978

Approved for public release; distribution unlimited.

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

78 14 08 011

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

18 1. REPORT NUMBER: AFGL-TR-78-0081

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6 4. TITLE (and Subtitle): FASCODE - FAST ATMOSPHERIC SIGNATURE CODE (Spectral Transmittance and Radiance)

5. TYPE OF REPORT & PERIOD COVERED: Scientific [redacted] 2

6. PERFORMING ORG. REPORT NUMBER: 14 VI-409

10 7. AUTHOR(s): H.J.P. Smith, D.J. Dube, M.E. Gardner, S.A. Clough†, F.X. Kneizys†, L.S. Rothman†

8. CONTRACT OR GRANT NUMBER(s): 15 F19628-77-C-0041

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Visidyne, Inc., 19 Third Avenue, Burlington, Massachusetts 01803

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 16 62101F 7670[redacted]; 61102F 2310G103

11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Geophysics Laboratory (OPI), Hanscom AFB, Massachusetts 01731, Contract Monitor: Francis X. Kneizys/OPI

12. REPORT DATE: 11 16 Jan[redacted] 1978

13. NUMBER OF PAGES: 149

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 12 142p. 17 06, G1

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

†Air Force Geophysics Laboratory (OPI)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Atmospheric transmission, equilibrium radiation, line-by-line calculations, line shape profiles, Voigt profile, Lorentz profile, Doppler profile, convolution techniques

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Using the HIRACC algorithm ~~developed at AFGL,~~ a rapid running line-by-line transmittance code has been developed for a layered atmosphere. In addition, the atmospheric radiance is also calculated assuming local thermodynamic equilibrium. The Voigt line shape profile is used for the general case but both the Lorentz and Doppler profiles may be chosen for cases where the Voigt shape is not needed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390 862 78 14 08 011 LB

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

1.0 INTRODUCTION

The problem of atmospheric transmission of infrared and visible radiation is of considerable interest from both a purely scientific and a practical point of view. The study of radiative transfer in an atmosphere involves a large number of physical and chemical effects and their proper theoretical treatment requires an increasing degree of mathematical sophistication. The vast amount of physical data necessary for a detailed description is typified by the AFGL line atlas which forms the core of the AFGL HITRAN Research Program[1]. Uses of this data include laboratory studies of absorption, computing data for comparison or design of field observations of the atmosphere as well as such applications as the design of remote sensing or laser communication systems.

In the design and simulation of such systems, the atmospheric transmission in a given wave number interval can be a crucial consideration. For systems with relatively coarse spectral resolution, the detailed spectral structure of the absorption is not required. Thus, for such systems a high resolution technique for study and simulation may not be necessary and one may profitably use lower resolution codes such as the AFGL LOWTRAN Program[2], but even in applied work a high degree of spectral resolution is increasingly in demand.

A number of line-by-line calculational methods have been reported which provide as detailed a spectral resolution as computationally possible. Among these may be found the LBL Program which has come to be know as the HITRAN Code[1]. This method uses the AFGL line atlas and convolves all lines contributing to a given wave number to within a prescribed wave number range using a Lorentz line shape profile. The large number of calculations which the LBL Program performs requires a large amount of computer time which makes parametric studies as well as wide-band computations prohibitively expensive. Thus this code tends to be used for very necessary high resolution work and also as a tool to improve lower resolution models such as LOWTRAN.

With the growing demand for high resolution work, an effort was undertaken at AFGL to improve the efficiency of line-by-line calculations. The initial phase of this work resulted in the development of the HIRACC algorithm for the convolution of Lorentz line shapes within 48 half-widths of a given wave number for a constant atmospheric temperature, density and composition. This extremely efficient algorithm resulted in a decreased running time of somewhat more than an order of magnitude as compared to the corresponding parts of the LBL Code. The HIRACC algorithm has been fully documented in an earlier report[3] and we shall assume in the current report that the reader is familiar with Reference [3].

At this point it would be helpful to the reader to summarize the main features of the HIRACC algorithm which combine to give effectively an order of magnitude decrease in computational time. The first step in the development of HIRACC involved the decomposition of the line shape profile into three sub-profiles with different spectral widths. The algorithm was developed for a pure Lorentz line shape and the line shape profile was cut off beyond 48 half-widths from the line center. FASCODE (Fast Atmospheric Signature Code), however, was designed to use a Voigt profile. In order to accomplish this it was found convenient to extend the cut-off of the profiles to 64 half-widths. This enabled the definition of only one new sub-profile, a Gaussian function representing the Doppler contribution out to 4 half-widths. Note that beyond this number of half-widths the Doppler broadening may be neglected. The Voigt profile is then approximated by an appropriate combination of four functions. Figures 1.1 and 1.2 illustrate the resulting decomposition.

This decomposition has the advantage that the rapid variation of the profile near the line center can be represented properly by a specified sampling interval and the more slowly varying portion as one approaches the wings of the line can be properly represented with a larger increment between the points. In addition, the results of the four separate convolutions are not put together until the contributions from all the lines in the region are complete. An interpolation scheme was developed to compute those values required between the numerical values. Details are given in Section 2.

Another important result of Reference [3] was the determination of a criterion for the optimum sampling interval. A sampling interval which gives results to ~ 0.1 percent was shown to be given by

$$\Delta\nu \simeq \alpha/4 \qquad (1.1)$$

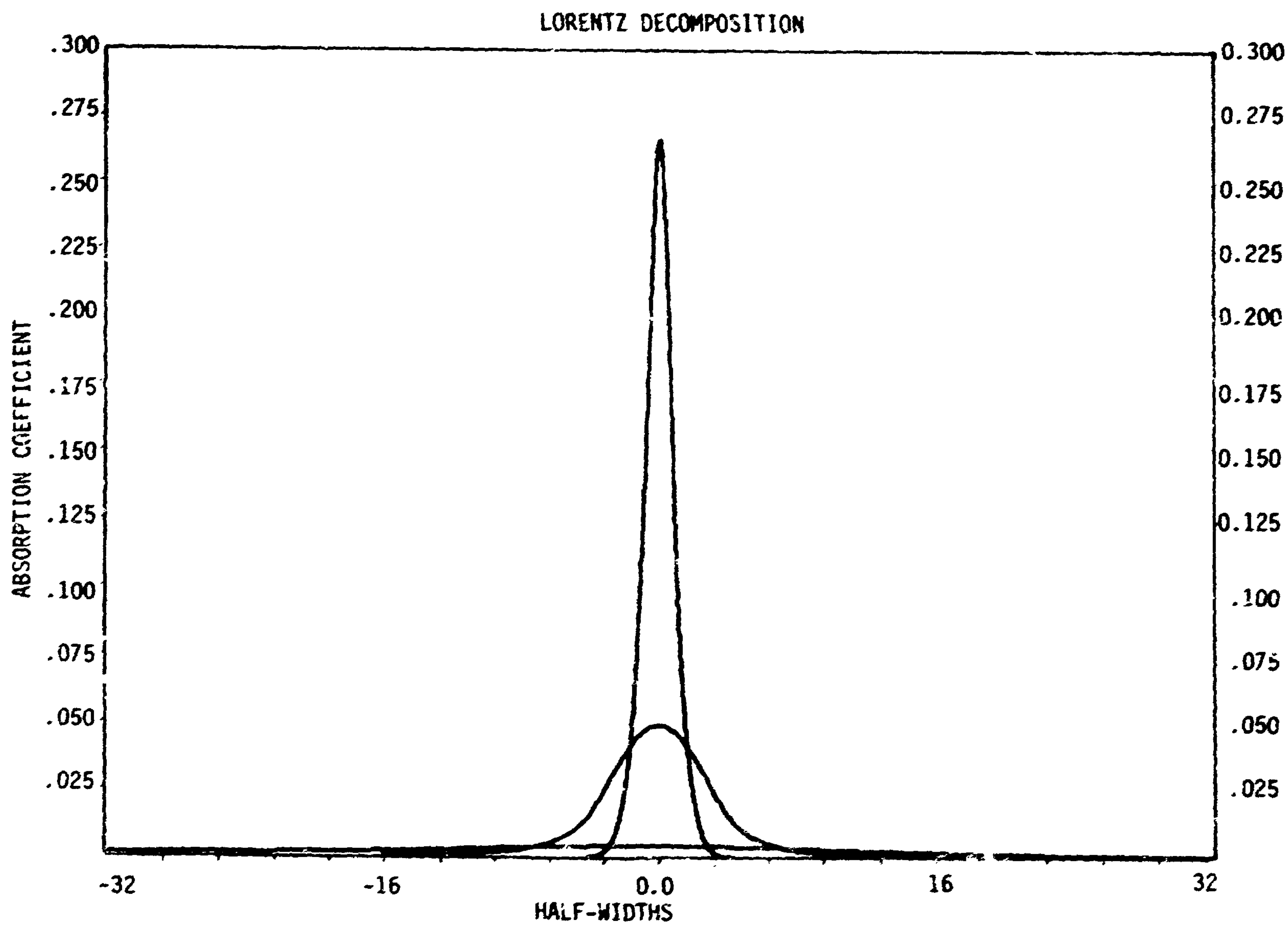

Figure 1.1: Decomposition of the Lorentz Function into Three Domains: -4 to 4 Half-Widths, -16 to 16 Half-Widths and -64 to 64 Half-Widths. The Sum of The Three Functions Gives $L(z) = (1/\pi)(1/1 + z^2)$. Only values to ± 32 half-widths are shown in the figure.

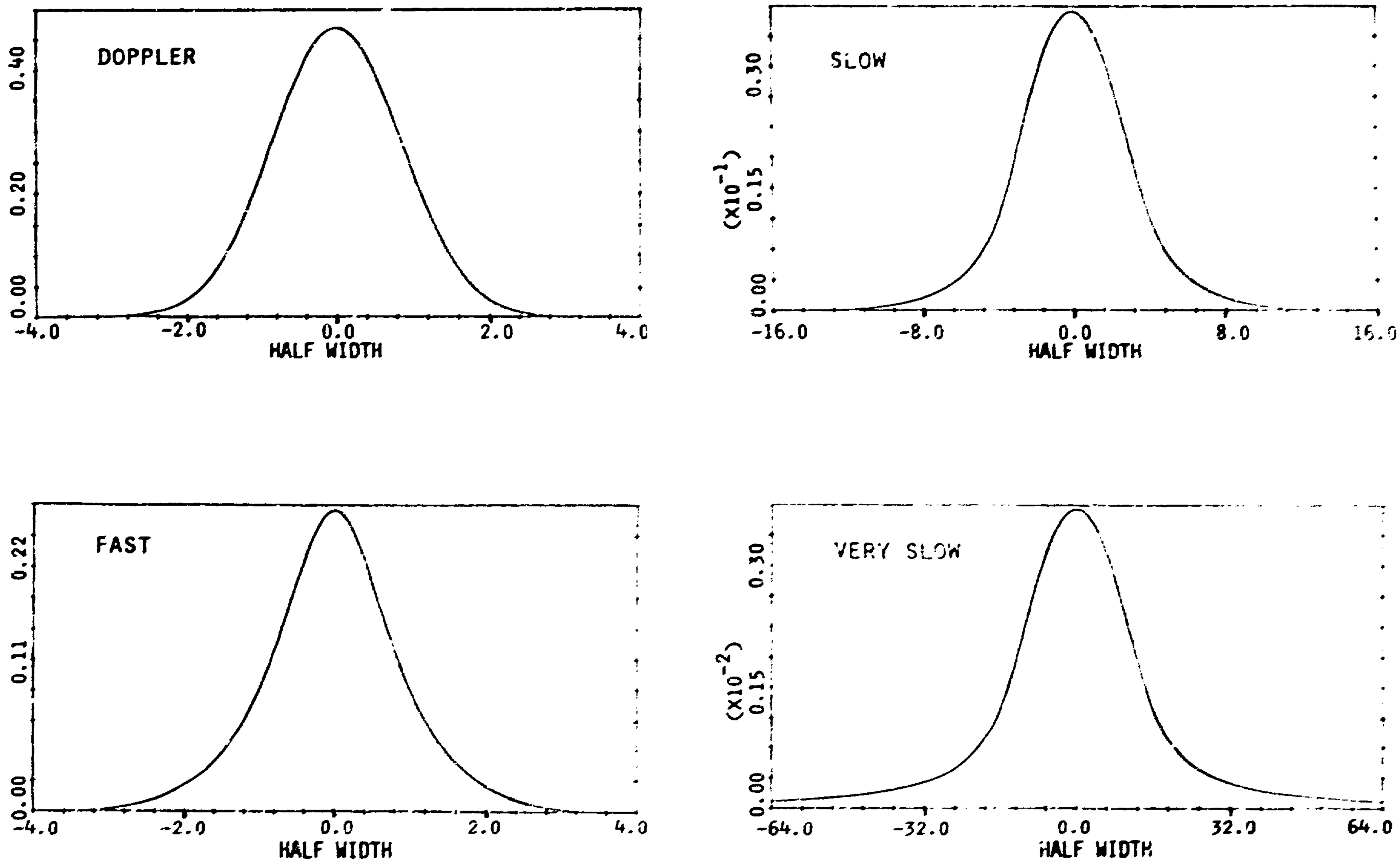

Figure 1.2: Four functions used to reconstruct the Voigt function.

where α is the average half-width of a line over the region of interest. It was also shown that the gain G obtained by the decomposition method over computing the entire line shape at a given sampling interval is given by the relation

$$G = N_{TOT} \Bigg/ \left[\sum_{m=1}^{M} N_M \, (\delta x_1/\delta x_m) \right] \tag{1.2}$$

In this expression, N_{TOT} is the total number of half-widths over which the profile is required, N_m is the number of half-widths spanned by the m^{th} decomposed function and $(\delta x_1/\delta x_m)$ is the ratio of the sampling interval for the first function to that required for the m^{th} function. For the decomposition utilized in this report, this yields a theoretical gain of 5 1/3.

We also point out another feature, namely that for a given spectral interval $(\nu_1 - \nu_2)$ which contains N lines, the same number of operations per line are performed for the sampling interval chosen according to the criterion. This implies that the convolution of a spectral absorptance for a given spectral interval will always take the same amount of computational time per line for high altitude layers where the lines are narrow as it does for lower layers where the lines may be broadened considerably. This feature is borne out by our results.

The HIRACC algorithm has been used to develop a multilayer transmission and radiance code which has been called FASCODE standing for Fast Atmospheric Signature Code. In the course of producing this code the basic HIRACC algorithm has been left intact, but several peripheral modifications have been implemented. In order to provide the capability of calculating at higher altitudes where the Lorentz profile is no longer appropriate, a Voigt line shape version has been developed with the additional feature that calculations are performed out to 64 half-widths. One of the great advantages of the HIRACC technique is that the line profile may be changed readily with only a small change in running time. The Voigt line shape profile is described in Section 2 of this report. We note in passing that purely Lorentz or purely Doppler versions of FASCODE may easily be implemented for the user whose application is in these domains. We emphasize, however, that no real penalty in running time will be paid by a user who exercises the Voigt version in either of these altitude domains.

As mentioned above, the HIRACC algorithm for absorption was developed for a uniform atmospheric path (temperature, pressure and absorber concentration). The application of this method to the real atmosphere was made by approximating the real atmosphere by a series of layers with constant parameters in each layer. The results for each layer are then merged with the absorption coefficient obtained up to that layer. The merging algorithms are described in detail in Section 3 of this report.

Assuming local thermodynamic equilibrium, the absorption of radiation by a given volume of gas implies the re-emission of an equal amount of energy. Using this idea one may "invert" a transmission calculation to obtain the radiance from the gas along the given path. This type of calculation has been provided as an option in FASCODE to enable the user to calculate the radiance due to the atmsophere itself. For the user who has a boundary at one end of his path, a provision has been made to allow the user to add his own surface radiance model. Currently this takes the form of a black body radiating at a given temperature at the low end of a path starting in space and looking down. This could be easily modified if a different spectral radiance is required. This part of FASCODE is described in Section 4.

In Section 5, we present some avenues for future efforts and summarize the status of this effort. A series of appendices gives an overview of the FASCODE structure as well as complete documentation of the new routines involved. One of these appendices provides a User's Manual for some sample inputs and outputs.

2.0 LINE SHAPE — VOIGT PROFILE

To calculate spectral absorption contours for the types of absorbing paths encountered in the atmosphere, extensive consideration must be given to the spectral line shape. For atmospheric paths at lower altitudes, the spectral line shape within ~5 cm^{-1} of the line center is given satisfactorily by the pressure broadened (Lorentz) shape. Beyond ~5 cm^{-1} it has been shown that the shape is dependent on molecular type as has been discussed by Burch[4], Winters et al[5] and Clough et al[3]. At high altitudes (low pressure) the line shape is due to thermal motion resulting in the Doppler line profile. The altitude above which the Doppler shape is valid is dependent on the wave number of the transition, the temperature and the molecular species. Between the domains for which the Lorentz shape and the Doppler shape are valid, the Voigt line shape must be utilized. Although the Voigt profile was originally derived for the case where broadening was due to natural broadening and thermal motion broadening, it is applicable in the present situation, since the shape function is the same for natural as for pressure broadening. This development uses an approximate method for the Voigt profile that is sufficiently accurate for most atmospheric problems and requires substantially reduced computer time compared to previously described methods.

The absorption coefficient for a pressure broadened line, $A_L(\nu)$, as a function of the wave number value of the radiation field, ν, the transition wave number, ν_0, intensity S and half-width, α_L, (half-width at half maximum: HWHM) is given by the Lorentz function,

$$A_L(\nu) = \frac{S}{\pi} \frac{\alpha_L}{\alpha_L^2 + (\nu - \nu_0)^2} \tag{2.1}$$

The pressure broadened half-width, α_L, is a function of the absorbing and broadening molecule types and the vibrational-rotational states involved in the transition. The theory that has proven useful for calculating pressure broadened half-widths is due to Anderson[6], as implemented by Tsao and Curnutte[7]. The variation of the half-width with temperature and pressure is given by the relationship

$$\alpha_L (P,T) = \alpha_L (P_o, T_o) \left(\frac{P}{P_o}\right)\left(\frac{T_o}{T}\right)^{X_T} \tag{2.2}$$

The half-widths on the AFGL line compilation are for P_o = 1 atm, T_o = 296°K and are typically of the order of 0.08 cm^{-1}/atm. The simplest theoretical result for the temperature dependence of the half-width gives X_T = 0.5. Recent calculations and experiments indicate that a higher value of X_T is more appropriate, $\lambda_T \simeq 0.75$ (Benedict)[8]. The absorption coefficient for a line broadened by thermal motion is given by the Gaussian function in terms of the Doppler width, α_D, (HWHM) as

$$A_D (\nu) = \sqrt{\frac{\ln 2}{\pi}} \; \frac{S}{\alpha_D} \exp\left[- (\ln 2) \left(\frac{\nu - \nu_o}{\alpha_D}\right)^2\right] \tag{2.3}$$

where

$$\alpha_D = \frac{\nu_o}{c} \left(\frac{2 \ln 2 \; kT}{(M/N_o)}\right)^{1/2} \tag{2.4}$$

For the half-width, α_D, to be given in cm^{-1} at temperature T(°K), c is the velocity of light in cm/sec, k is the Boltzmann factor in erg/deg, N_o is Avogadro's number, and M is the molecular weight of the molecule type in gr.

Both functions have been defined such that the functional value at one half-width from the line center is one-half the function value at the line center, that is

$$\frac{A (\nu_o \pm \alpha)}{A(\nu_o)} = \frac{1}{2} \tag{2.5}$$

It is also important that the integral of both functions yields the line strength, that is

$$S = \int_{-\infty}^{\infty} A(\nu)\, d\nu \tag{2.6}$$

For the purposes of this development, and in order to utilize the algorithm developed by Clough et al[3], it will be useful to define a dimensionless argument for the line profiles,

$$z = \frac{\nu - \nu_0}{\alpha} \tag{2.7}$$

so that the Lorentz profile is given by

$$A_L(z) = \frac{1}{\pi} \frac{S}{\alpha_L} \frac{1}{1 + z^2} \quad \text{with } z \equiv \frac{\nu - \nu_0}{\alpha_L} \tag{2.8}$$

and the Doppler profile is given by

$$A_D(z) = \sqrt{\frac{\ln 2}{\pi}} \frac{S}{\alpha_D} \exp\left[-(\ln 2)\, z^2\right] \quad \text{with } z \equiv \frac{\nu - \nu_0}{\alpha_D} \tag{2.9}$$

The integral and half-width properties take on the following form

$$S = \int_{-\infty}^{\infty} A(z)\ \alpha dz \tag{2.10}$$

and

$$\frac{A\left(z = \frac{1}{2}\right)}{A(z = 0)} = \frac{1}{2} \tag{2.11}$$

Several excellent programs have been written to calculate the Voigt line profile which may be regarded as the convolution of the Lorentz function with the Gaussian function (J.H. Pierluissi, P.C. Vanderwood and R.B. Gomez[9]; S.R. Drayson[10] and B.H. Armstrong[11]). For the application of performing line-by-line calculations for a multi-layered atmosphere involving large numbers of spectral lines, these methods require a prohibitive amount of computational time and provide more accuracy than is generally required for such problems. Approximations for the computation of the Voigt profile have been suggested by Whiting[12] and Kielkopf[13]. The method described in the latter two papers involves approximating the Voigt function by a weighted sum of the Doppler and Lorentz functions. It is an extension of this approach that we have utilized in this development.

The startingpoint for our approximation to the Voigt function is the definition of a Voigt parameter that is simply calculable and is well behaved in the Lorentz and Doppler limits. For a spectral transition with a Doppler width, α_D, and a Lorentz width, α_L, we define a Voigt parameter, ζ, where

$$\zeta = \frac{\alpha_L}{\alpha_L + \alpha_D} \tag{2.12}$$

In the Lorentz and Doppler limits respectively we have,

$$\zeta = 0 \qquad \alpha_L << \alpha_D \qquad \text{(Doppler Limit)} \tag{2.13}$$

and

$$\zeta = 1 \qquad \alpha_L >> \alpha_D \qquad \text{(Lorentz Limit)} \tag{2.14}$$

The definition of a Voigt half-width, α_V, is now required to proceed with our program. This definition is made consistent with the definition of the half-width for the Doppler and Lorentz functions; it is the half-width at half maximum of the Voigt function. An excellent approximation has been given by Kielkopf in terms of the Lorentz and Doppler widths,

$$\alpha_V = \frac{\alpha_L}{2}(1 + \varepsilon) + \left[\frac{\alpha_L^2 (1 - \varepsilon)^2}{4} + \alpha_D^2\right]^{1/2} \qquad (2.15)$$

with $\varepsilon = 0.0990 \ln 2$.

In order to utilize this expression in terms of ζ, it is necessary to consider the determination of α_V in two domains of ζ:

$$\alpha_V = a_{VD}(\zeta) \cdot \alpha_D \qquad 0.0 \leq \zeta \leq 0.5 \quad \text{(Doppler Regime)} \qquad (2.16)$$

and

$$\alpha_V = a_{VL}(\zeta) \cdot \alpha_L \qquad 0.5 \leq \zeta \leq 1.0 \quad \text{(Lorentz Regime)} \qquad (2.17)$$

The quantities, $a_{VD}(\zeta)$, and $a_{VL}(\zeta)$, may be obtained from Equation (2.15) in terms of ζ as

$$a_{VD}(\zeta) = \frac{\alpha_V}{\alpha_D} = \left(\frac{1 + \varepsilon}{2}\right)\left(\frac{\zeta}{1 - \zeta}\right) + \left[\left(\frac{1 - \varepsilon}{2}\right)^2 \left(\frac{\zeta}{1 - \zeta}\right)^2 + 1\right]^{1/2}$$

$$0. \leq \zeta \leq 0.5 \qquad (2.18)$$

and

$$a_{VL}(\zeta) = \frac{\alpha_V}{\alpha_L} = \left(\frac{1 + \varepsilon}{2}\right) + \left[\left(\frac{1 - \varepsilon}{2}\right)^2 + \left(\frac{1 - \zeta}{\zeta}\right)^2\right]^{1/2}$$

$$0.5 \leq \zeta \leq 1.0 \qquad (2.19)$$

The largest error in the determination of the Voigt half-width using Equation (2.15) is $\simeq 0.02\%$. Values of a_{VD} with $0.0 \leq \zeta \leq 0.5$ and for $a_{VL}(\zeta)$ with $0.5 \leq \zeta \leq 1.$ using Equations (2.18) and (2.19) are determined for equally spaced values of ζ separated by 0.005. Plots of a_{VD} and a_{VL} as a function ζ appear in Figure 2.1 and Figure 2.2 respectively. This error is not as great as that incurred in using values of a_{VD} and a_{VL} at discrete values of ζ for

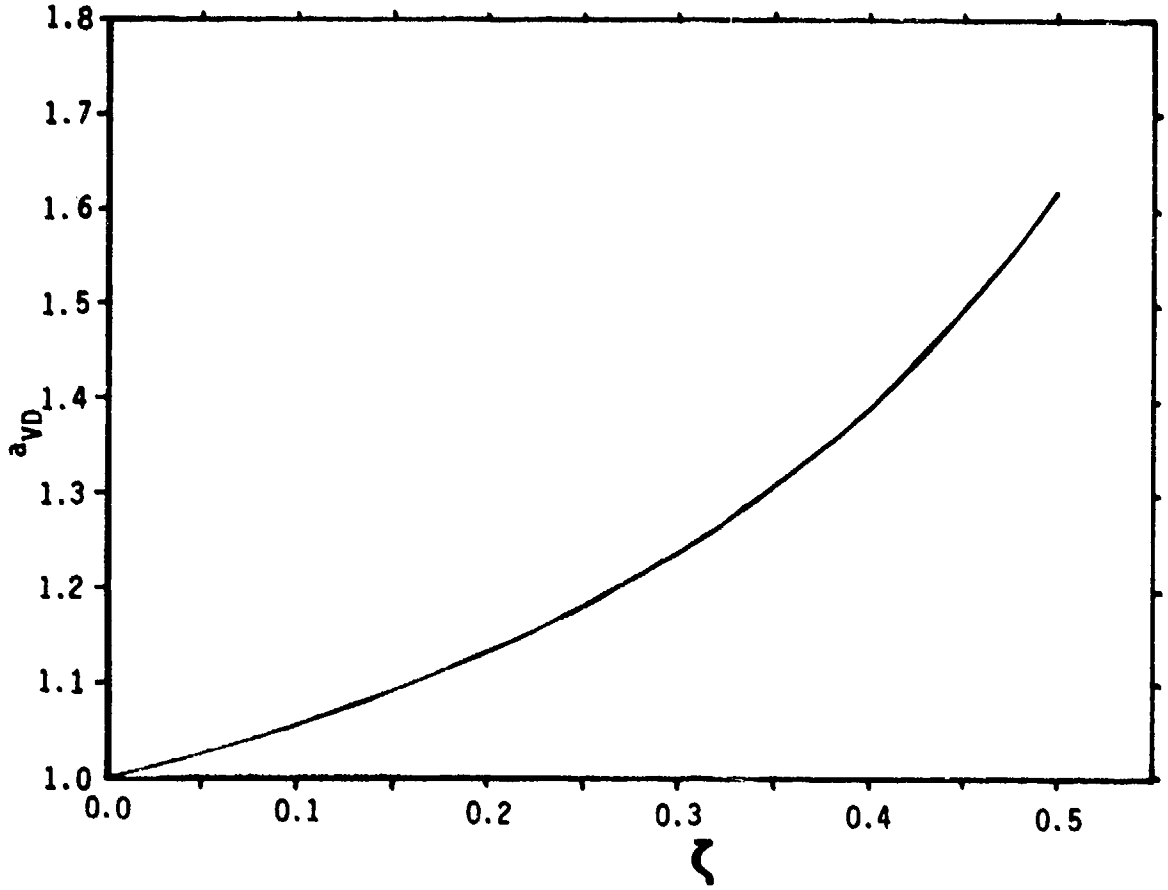

Figure 2.1: Plot of a_{VD} Vs. ζ ($0.0 \leq \zeta \leq 0.5$)

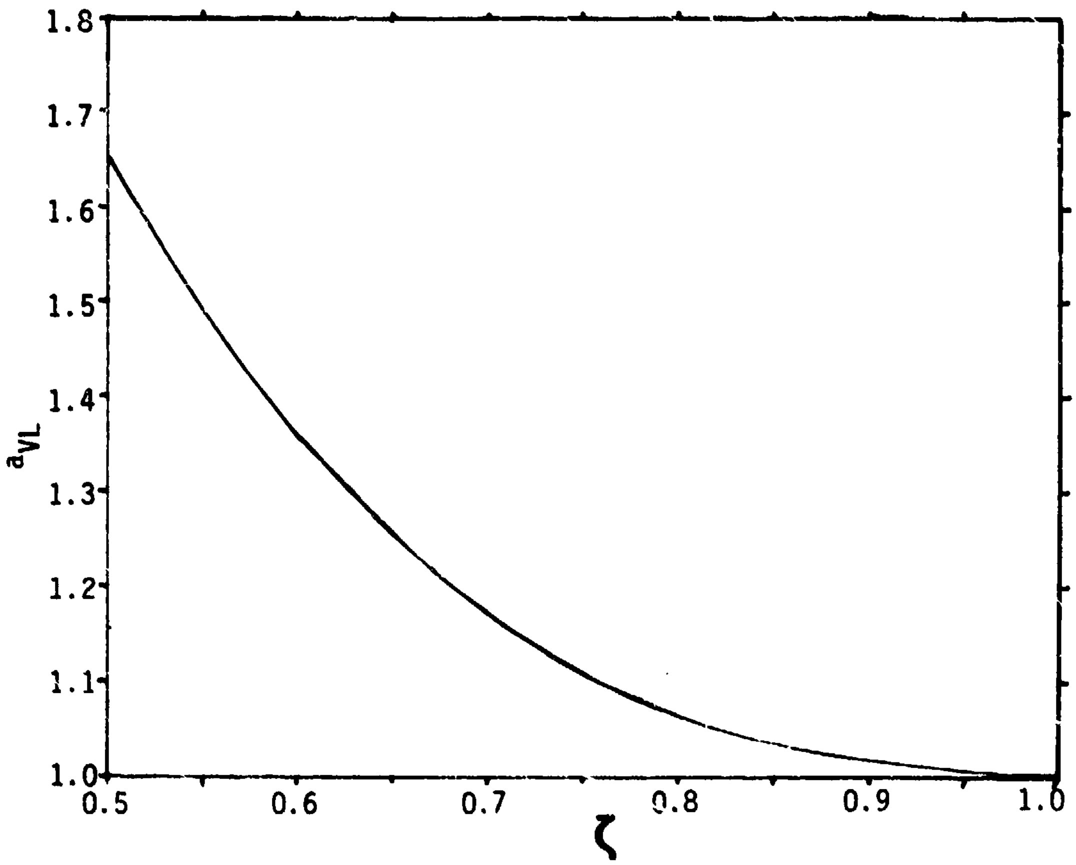

Figure 2.2: Plot of a_{VL} Vs. ζ $(0.5 \leq \zeta \leq 1.0)$.

values of ζ intermediate to those tabulated. For most physical problems, the errors involved in the present method of determining α_V, are less than the errors in the values of α_D and α_L themselves. If it is desired to have more accurate values of α_{VD} and α_{VL}, the Voigt program of Armstrong can be utilized in an iterative procedure. In general, only one iteration is required to give a high degree of accuracy.

The Voigt function may now be approximated as a weighted sum of Doppler and Lorentz functions of width, α_V,

$$A_V(\zeta, \alpha_V, Z) = (1 - C(\zeta))\, A_D(\alpha_V, Z) + C(\zeta)\, A_L\, (\alpha_V, Z) \tag{2.20}$$

with $Z = (\nu - \nu_o)/\alpha_V$

This definition preserves the integral property desired: that the integral over Z gives back the line strength, S, since $C(\zeta)$ is independent of Z and the two functions, A_D and A_L, are themselves normalized to the line strength. Consider

$$\int_{-\infty}^{\infty} A_V(\zeta, \alpha_V, Z)\, dZ = (1 - C(\zeta)) \int_{-\infty}^{\infty} A_D(\alpha_V, Z)\, \alpha_V\, dZ$$

$$+ C(\zeta) \int_{-\infty}^{\infty} A_L\, (\alpha_V, Z)\, \alpha_V\, dZ, \tag{2.21}$$

and

$$= (1 - C(\zeta))\, S + C(\zeta) S \tag{2.22}$$

so that

$$= S \tag{2.23}$$

The remaining problem is the determination of the weighting constant, $C(\zeta)$.

In References [12] and [13] analytic functions have been given for the constant $C(\zeta)$ as a function of Voigt parameter. The difficulty with the procedure outlined there is the large number of operations required to attain $C(\zeta)$ from the analytic expressions given. The most straightforward way to proceed is to determine the numerical function $C(\zeta)$ for the same 201 values of ζ used to determine the Voigt width. Rather than using the analytic expressions given in References [12] and [13] a least squares procedure has been used to determine $C(\zeta)$ by minimizing the weighted sum of the deviations squared, σ^2, as obtained from Equation (2.20),

$$\sigma^2 = \sum_i W_i \left\{ C(\zeta) \left[A_L (\alpha_V, Z_i) - A_D (\alpha_V, Z_i) \right] - \left[A_V (\zeta, \alpha_V, Z_i) - A_D (\alpha_V, Z_i) \right] \right\}^2 \tag{2.24}$$

All the quantities have been defined except the weighting function W_i and the grid of points represented by Z_i. Some experience indicated that the weighting scheme which presented the best compromise between error in the central portion of the line profile and the line wing was obtained by setting the weight to the inverse of the Voigt value, that is

$$W_i = 1/A_V(\zeta, \alpha_V, Z_i) \tag{2.25}$$

A more obvious choice would have weighted the deviations inversely as the square of the Voigt value to maintain nearly constant percent error across the line profile. It was deemed more important to maintain a smaller percent error for the larger values of the function near the center of the line. The points were chosen at equally spaced intervals over three different domains for reasons that will be discussed later. The grid was chosen as follows:

$$i = 1, 19 \qquad Z_1 = 0,\ Z_2 = 0.25,\ \cdots\cdots,\ Z_{19} = 4.0$$

$$i = 20, 30 \qquad Z_{20} = 5.0,\ Z_{21} = 6.0,\ \cdots\cdots,\ Z_{30} = 16.0$$

$$i = 31, 41 \qquad Z_{31} = 20.,\ Z_{32} = 24.,\ \cdots\cdots,\ Z_{41} = 64.0$$

The results obtained for the constant, $C(\zeta)$, are shown in Figure 2.3 as a function of ζ. The values are determined at the same 201 values of ζ in Figures 2.4 through 2.8. The continuous curves are the result of the Armstrong Voigt program and the x's are the values obtained from the least squares procedure. For $\zeta = 0$ and $\zeta = 1$ the Voigt function is exactly reproduced as a required result of the method. The results are given on a logarithmic plot in order to give perspective to the value of the function for which the percent errors are the largest. For $\zeta = 0.05$ to $\zeta = 0.3$, the percent error in the wing is of the order of 22 percent, but these errors occur for very small values of the function. Another region of moderate percent error is at $Z \simeq 3$ half-widths for this same range of zeta. The largest percent error in this domain is ~ 8% for zeta $\simeq$ 0.2.

As indicated earlier, these results are sufficiently accurate for most atmospheric applications. As suggested by both Whiting[12] and Kielkopf[13], an error function can be added to reduce this remaining discrepancy to negligible proportions. Some discussion of this will appear in a subsequent section.

The method for approximating the Voigt profile has been developed and this development must now be incorporated into the algorithm for computing spectral absorption coefficients outlined by Clough et al[3]. The Voigt function has been obtained as a weighted sum of the Doppler function and the Lorentz function. For purposes of convolving the approximate Voigt profile with the spectral line data, we consider Equations (2.8) and (2.9). In Reference [3] the slow convergence of $A_L(z)$ has been discussed and a method was developed to reduce computational effort for performing the convolution. Fortunately, the Doppler shape has very rapid convergence to zero as a function of the argument, z. The domain of the argument, z, for which values of the function $A_D(z)$ need to be considered are limited to $|z|$ less

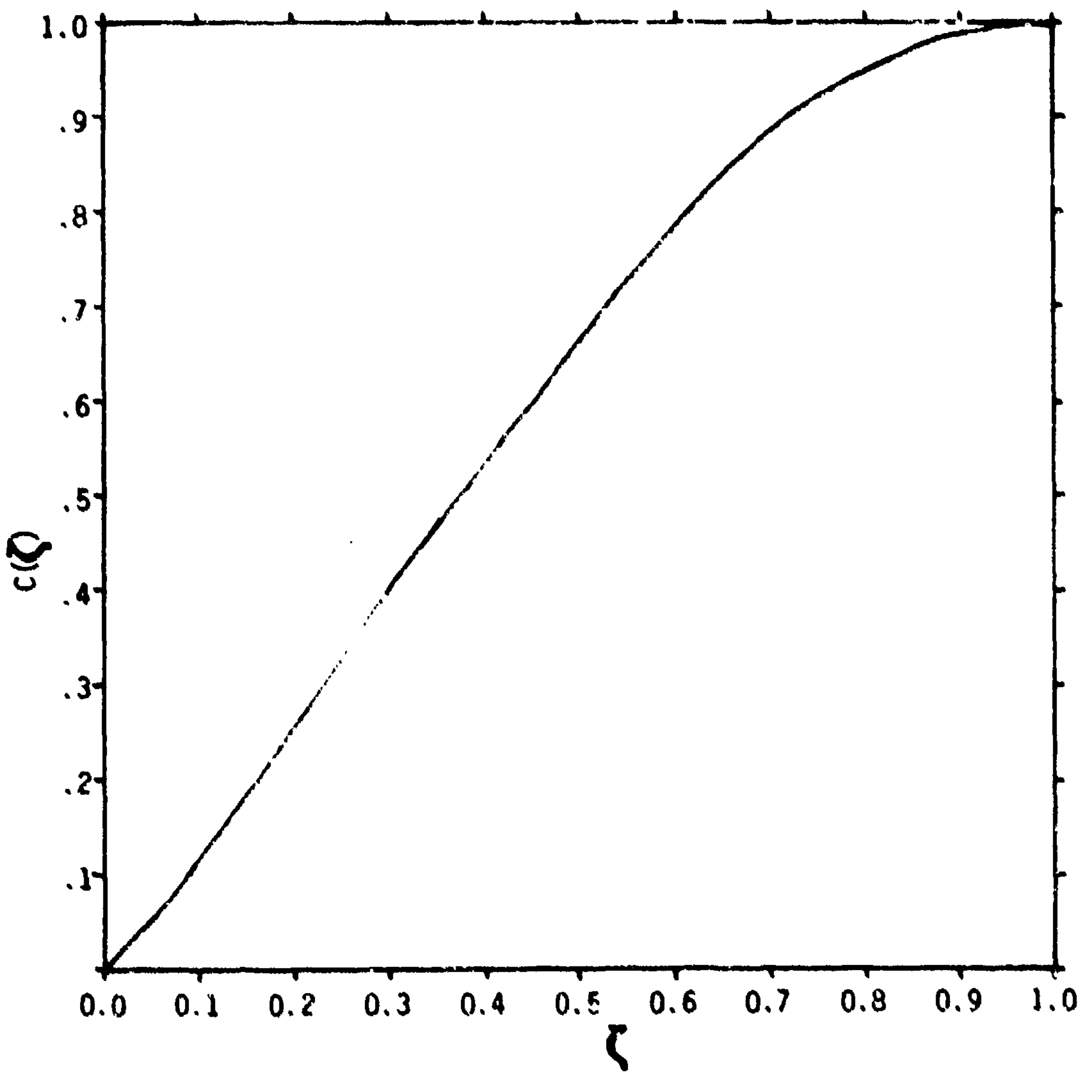

Figure 2.3: Plot of Least Squares Fit Parameter, C(ζ) Vs. ζ.

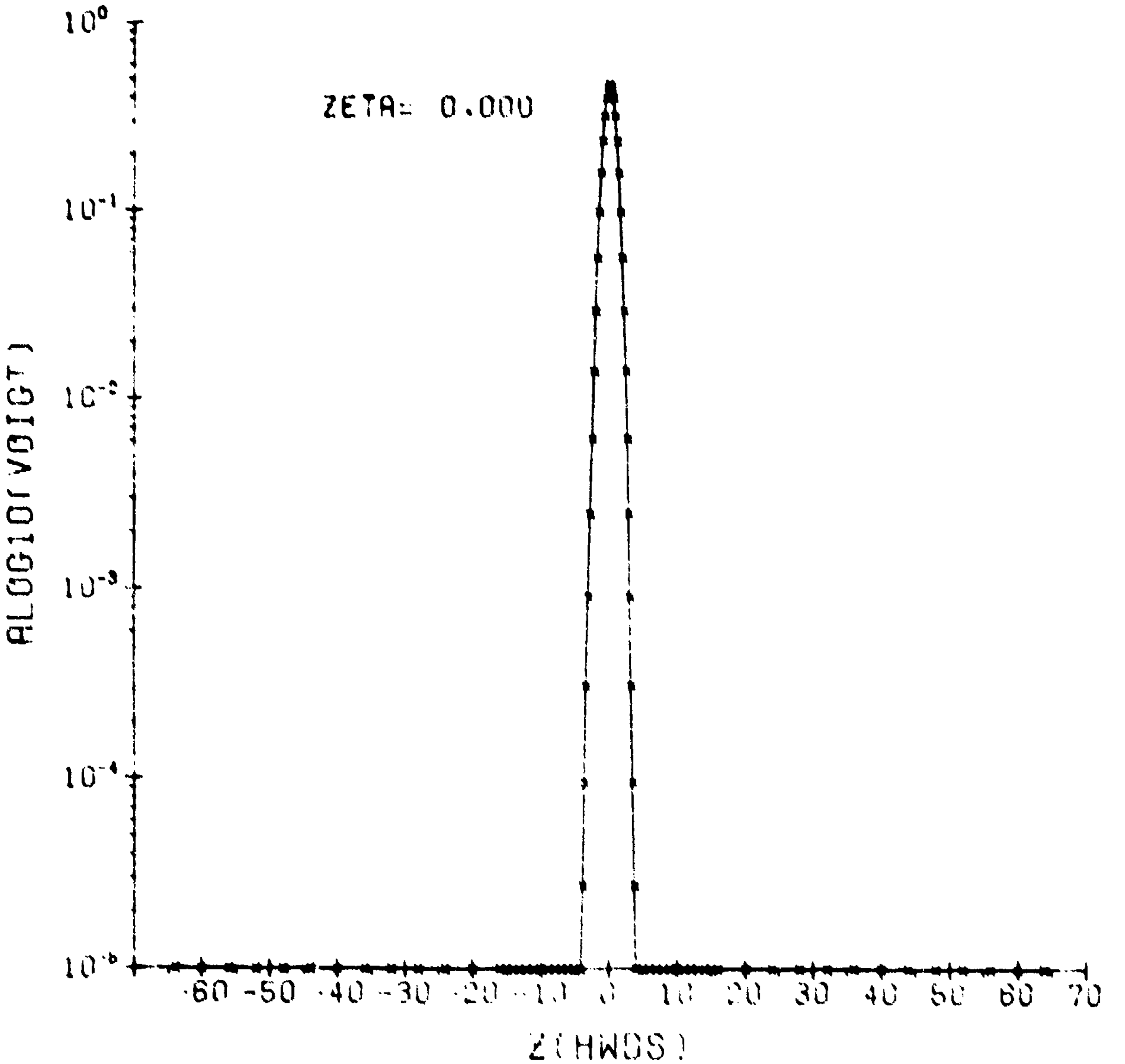

Figure 2.4: Voigt Line Shape Profile for ζ = 0.

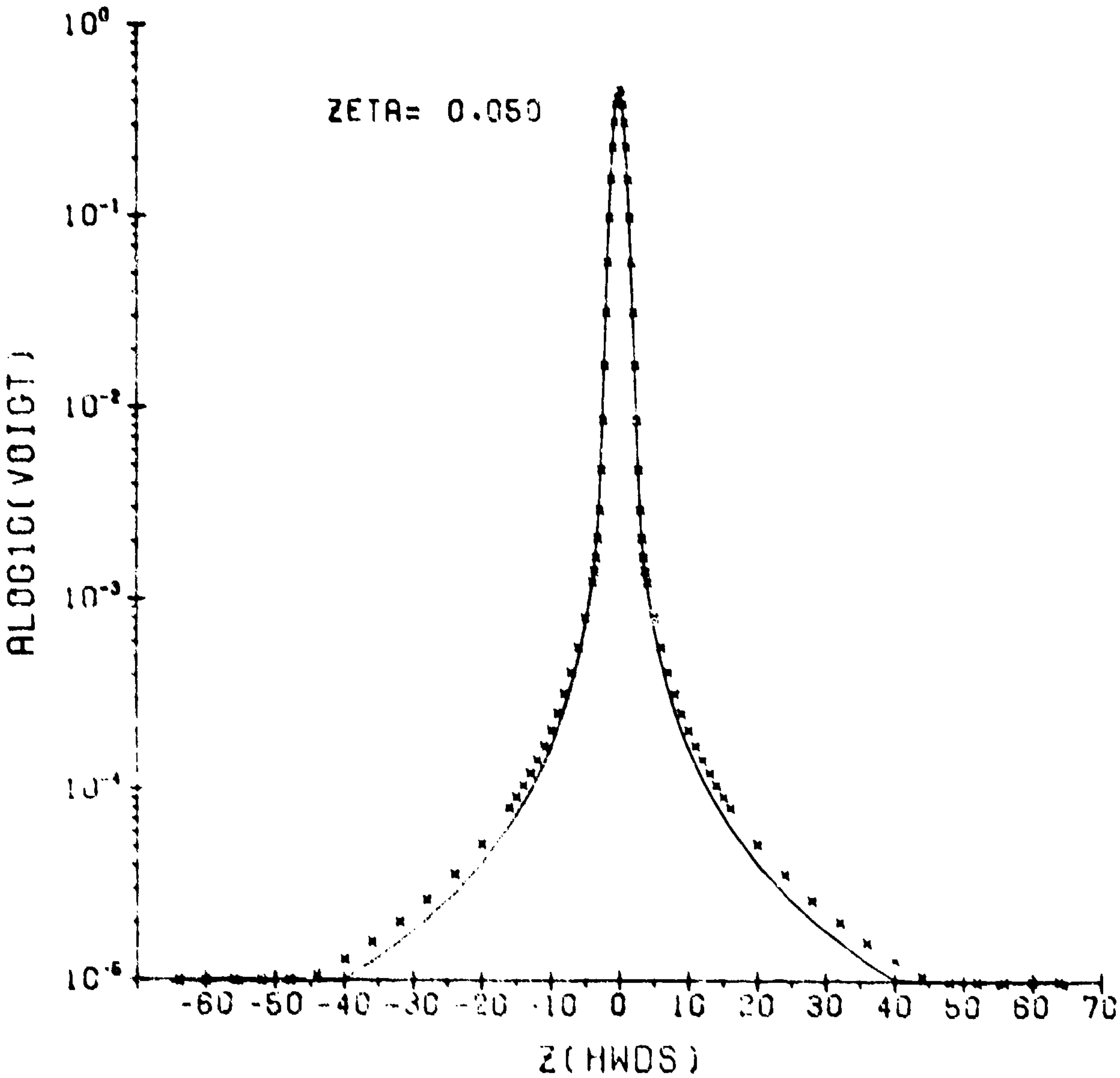

Figure 2.5: Voigt Line Shape Profile for ζ = 0.050

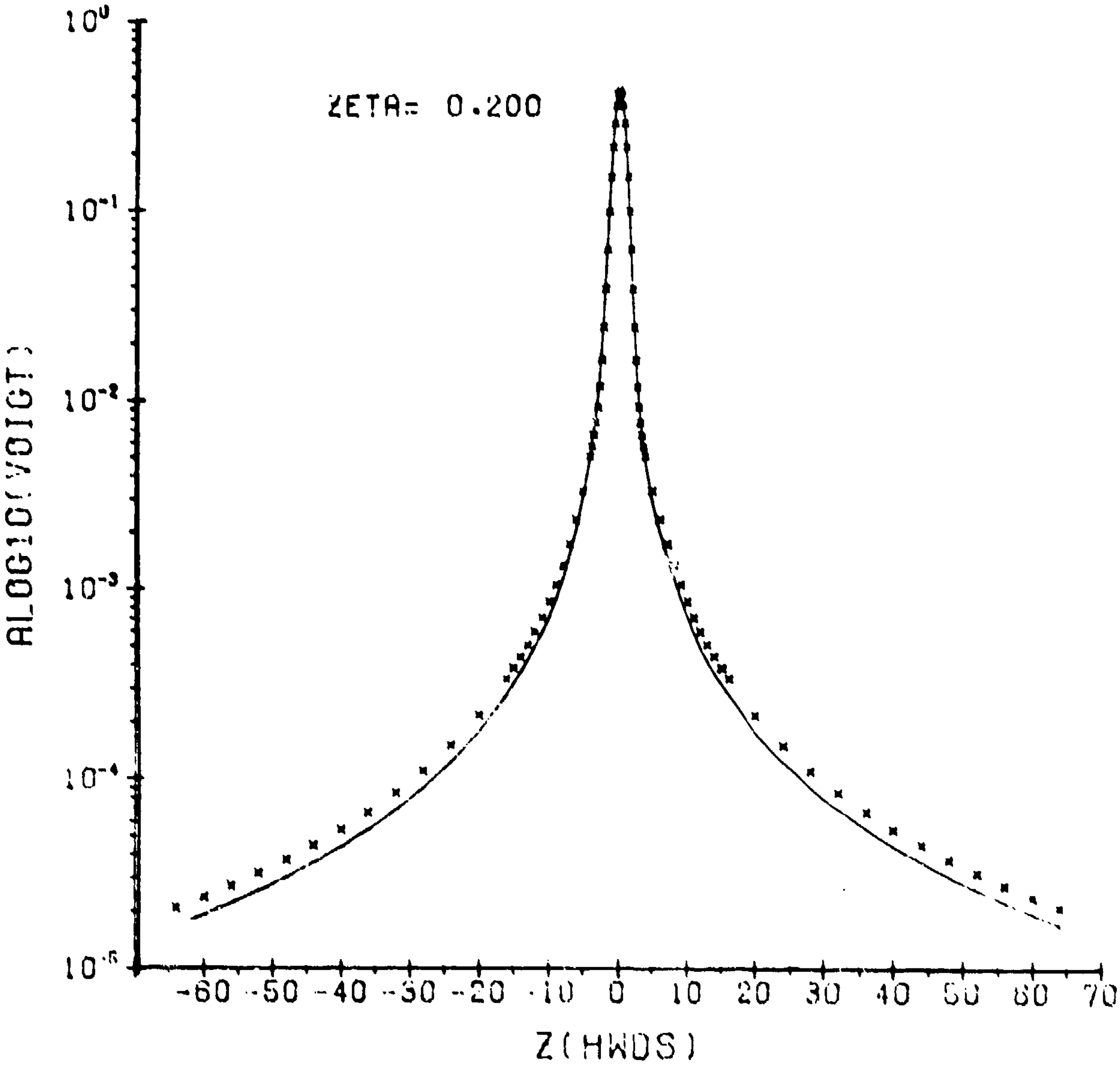

Figure 2.6: Voigt Line Shape Profile for ζ = 0.200.

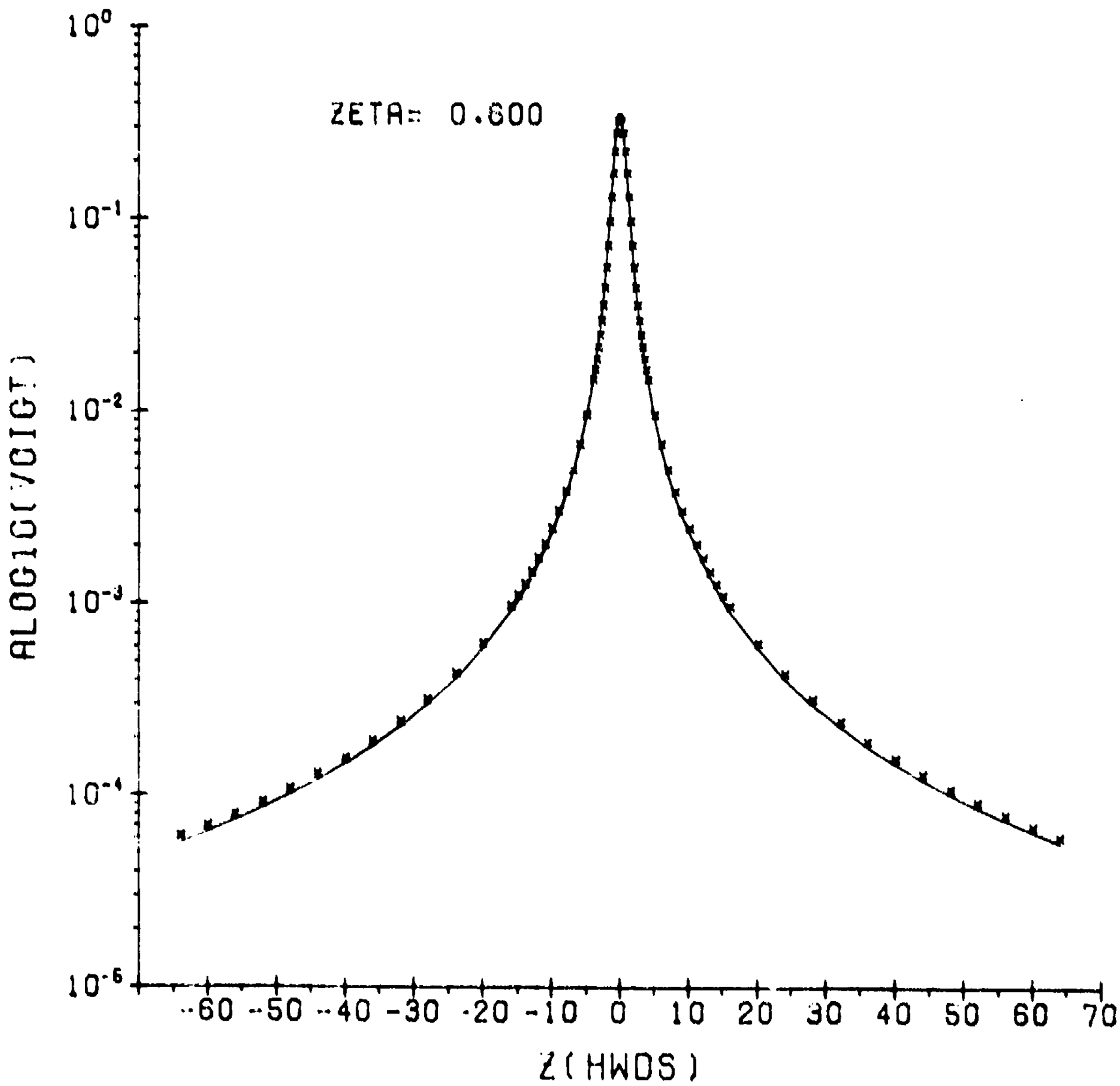

Figure 2.7: Voigt Line Shape Profile for ζ = 0.600.

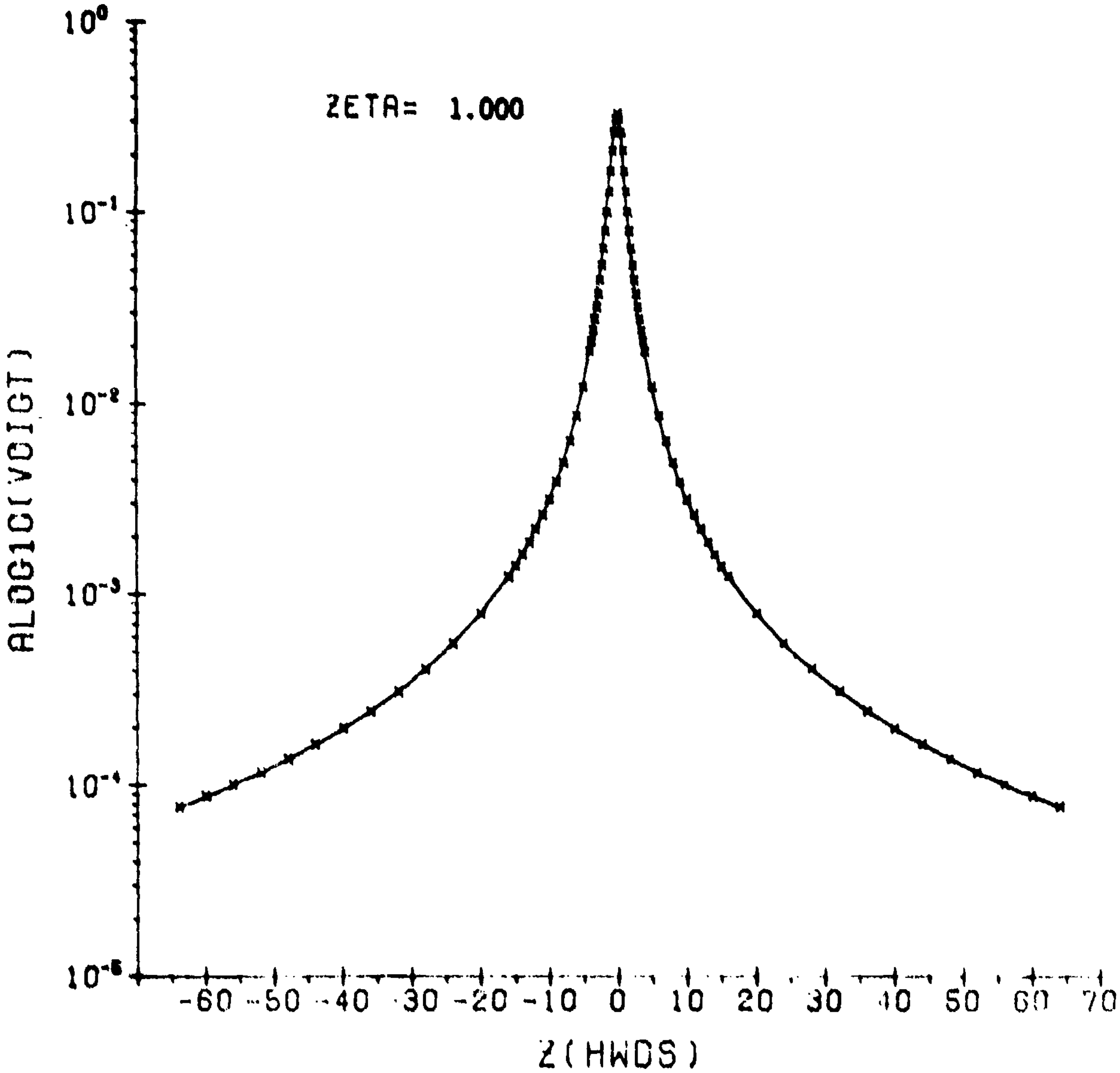

Figure 2.8: Voigt Line Shape Profile for ζ = 1.000.

than four ($|z| \leq 4$), noting that A_D $(z = 4) = 7.7 \times 10^{-6}$. The normalized function for the Doppler line profile is given by

$$G(z) = \sqrt{\frac{\ln 2}{\pi}} \exp\left[- (\ln 2)\ z^2\right] \tag{2.27}$$

It then is appropriate to decompose the Lorentz function $A_L(z)$ into three domains of z,

$$0 \leq z \leq 4$$

$$0 \leq z \leq 16$$

and

$$0 \leq z \leq 64 \tag{2.28}$$

This decomposition is performed in a manner similar to that described in Reference [3]. A quartic function is defined over the domain $0 \leq z \leq 4$ where

$$Q_1 = \frac{1}{\pi} (a_1 + b_1\ z^2 + c_1\ z^4) \tag{2.29}$$

The normalized Lorentz function, L, is

$$L = \frac{1}{\pi}\ \frac{1}{1 + z^2}\ . \tag{2.30}$$

The coefficients of the quartic are chosen such that for the function $(L - Q_1)$, the value and the first and the second derivative are zero at the boundary, Z_b. For the first function, $Z_b = Z_1 = 4$. These constraints are achieved by the following relations

$$a_1 = (1 + 3\ Z_b^2 + 3\ Z_b^4)/(1 + Z_b^2) \tag{2.31}$$

$$b_1 = -\ (1 + 3\ Z_b^2)/(1 + Z_b^2)^3 \tag{2.32}$$

and

$$c_1 = 1/(1 + Z_b^2)^3 \tag{2.33}$$

Similarly, a second quartic function, Q_2, is defined such that for function $(L - Q_2)$, the value, the first and second derivative are zero at the boundary $Z_b = Z_2$. In this case $Z_b = Z_2 = 16$. In the region from $16 \leq |Z| \leq 64$ the function utilized is the Lorentz function itself. The procedure is more obvious in tabular form:

		DOMAIN		
Function	$0 \leq \|Z\| \leq 4$		$0 \leq \|Z\| \leq 16$	$0 \leq \|Z\| \leq 64$
		(Doppler)		
$X_D(z)$	$G(z)$		0	0
		(Lorentz)		
$X_{L1}(z)$	$L(z) - Q_1(z)$		0	0
$X_{L2}(z)$	$Q_1(z) - Q_2(z)$		$L(z) - Q_2(z)$	0
$X_{L3}(z)$	$Q_2(z)$		$Q_2(z)$	$L(z)$

Note that the Doppler function spans the same domain as the first decomposed Lorentz function, that the functions X_{Li} sum to $L(z)$ in each domain and that the functions are continuous across the domain boundaries.

The four functions that will be utilized to reconstruct the Voigt function are shown in Figure 1.2. The functions are tabulated at 201 values of the argument $|z|$ over the domain valid for each function.

The total expression for approximating the absorption due to the Voigt profile is given by

$$A_V(\zeta, \alpha_V, z) = \frac{S}{\alpha_V}\left\{ \Big(1 - C(\zeta)\Big) X_D\ (z) \right. \tag{2.34}$$

$$\left. + C(\zeta) \Big[X_{L1}(z) + X_{L2}(z) + X_{L3}(z) \Big] \right\}$$

This may be separated into three functions where

$$A_{V1}(\zeta, \alpha_V, z) = \frac{S}{\alpha_V}\left\{ \Big(1 - C(\zeta)\Big) X_D(z) + C(\zeta)\ X_{L1}\ (z) \right\} \quad 0 \le |z| \le 4,$$

$$A_{V2}(\zeta, \alpha_V, z) = \frac{S}{\alpha_V}\ C(\zeta)\ X_{L2}\ (z) \qquad 0 \le |z| \le 16,$$

$$A_{V3}(\zeta, \alpha_V, z) = \frac{S}{\alpha_V}\ C(\zeta)\ X_{L3}(z) \qquad 0 \le |z| \le 64. \tag{2.35}$$

The sampling interval established in Reference [3] indicates that A_{V1} is sampled at intervals of $\frac{\alpha_V}{4}$, A_{V2} at intervals α_V, and A_{V3} at intervals of $4 \cdot \alpha_V$. This sampling scheme results in each function being sampled at 33 values. The results of the convolutions over the three domains are stored in arrays FF, SF and VSF. The saving in computational effort results from two principal reasons. Only 99 values are required to describe the Voigt function over $\pm$ 64 half-widths. The composite spectrum from the three arrays (FF, SF, and VSF) is constructed only after the convolutions have been completed for all the lines in a given spectral interval. This latter point is one that was not sufficiently stressed in Reference [3]. The implication of this technique is that not only is the grid of the function fine in the region where the function is varying rapidly and coarse in the region where the function is varying slowly (actually there are three discrete sampling intervals), but that the values for intermediate points are not calculated until after the contributions from all the spectral lines have been determined. This is achieved by interpolating the VSF array into the SF array, and the SF array into the FF array yielding the final results.

3.0 TRANSMISSION THROUGH THE REAL LAYERED ATMOSPHERE

The HIRACC algorithm[3] as originally reported by Clough et al computed absorption coefficients for a given path through a gas. The thermodynamic properties over this path were assumed to be constant which enabled the definition of a constant sampling interval over the spectral region of interest. The path itself was defined through input in the form of column densities of the appropriate molecular absorbers together with the temperature and pressure. We note that column densities are the required units for the AFGL tape.

At first glance, it might appear that calculations of the spectral absorptance (or optical depth) in the real atmosphere would require only the definition of the proper values of these column densities. However, one must take into account the variation of thermodynamic properties along the path as well as the fact that certain absorber molecules, such as H_2O and O_3 are not uniformly mixed at all altitudes. For example, ozone concentration peaks in the stratosphere. When a calculation of the optical depth at a higher altitude is required, one must take into account the fact that the decreased pressure implies a narrower line width. Indeed, as one goes from sea level toward space, the line shape profile passes from almost pure Lorentz to pure Doppler. If one uses an improper sampling interval to calculate the spectral absorptance, it is clear that important absorption features can easily be neglected, thus yielding incorrect results. Therefore a proper calculation of absorption over a path traversing a large region of the atmosphere will require an appropriate variation of the sampling interval with altitude. One could, of course, use the small sampling interval required at high altitude layers for all layers, but this would require unnecessary calculations at the lower altitudes.

In order to use the HIRACC algorithm with as little modification as possible, we have taken the approach discussed by McClatchey et al[14] and approximated the real atmosphere by a series of layers, each defined to have constant pressure and temperature and appropriate values of the column densities of the absorbing molecules. Clearly such a decomposition is not unique and requires the user to exercise some care in defining the atmosphere as will be discussed below. While this arrangement may demand a certain degree of sophistication of the user, it is judged that this solution will enable a user to be more flexible in changing from one program to another than he would be if the program were tied to a given atmospheric model.

The basic structure of FASCODE is then a successive application of the HIRACC algorithm for successive atmospheric layers with an appropriate merging of the results. For convenience, the HIRACC Program reported in Reference [3] has been left almost unchanged except to modify it to subroutine form and to include the Voigt line shape profile described in the previous section.

The merger of the absorptance for two different layers is performed in the following manner. The results for the first layer are computed and written to disk as described in Reference [3]. (We remind the reader that the HIRACC algorithm processes the spectral absorptance in "panels", namely in groups of data defined at wave numbers separated by a sampling interval. Typically there are 2400 quantities in a panel but a panel may be shorter at the end and beginning of the requested wave number interval.) The resolution at which the first layer is to be calculated is determined by the sampling interval criterion described by Clough et al[3], namely one-quarter of the average half-width of the lines. The next layer to be computed may have pressure and temperature sufficiently different such that a new sampling interval is required by the sampling criterion. Let the sampling interval for the first layer be DV1 and that for the second, DV2. In the initial development of FASCODE a decision was made to limit the program to specified ratios DV1/DV2. Note that since the pressure in the atmosphere is a monotonically decreasing function of altitude, the average half-width of a typical Lorentz line profile will decrease as the calculation proceeds from the lower to higher altitudes. Thus the sampling interval determined using the criterion of Reference [3] will also decrease as the calculation proceeds to the higher altitudes, until the pressure no longer remains the principal determining factor. This will occur at altitudes where the Doppler width becomes dominant in the determination of the half-widths. Such effects are already included in the Voigt line shape profile model discussed above. In addition, the Voigt model provides a proper treatment of the transition region at those altitudes where neither the Doppler nor the Lorentz profile is appropriate.

In general then, the sampling interval is larger at lower altitudes and smaller at the higher levels. This implies that a calculation which proceeds from the lower to the higher layers is more efficient since one would not have to perform the calculation of the lower layers at the small sampling intervals required by the narrower line shapes at the higher altitudes. Therefore, in FASCODE, the HIRACC algorithm is implemented by starting at the lowest layer (highest pressure) and proceeding to higher layers (lower pressures).

Since the decision was made to limit the number of ratios of the sampling intervals between two adjacent layers, FASCODE resets the DV of a new layer to the nearest allowed ratio. Note that the user must be careful to prepare the atmospheric input with a sufficient number of layers so that the criterion for the ratios is met. This is not a difficult condition to meet and it is discussed in detail in the User's Manual in Appendix B. In principle, the restriction to a fixed number of ratios is not necessary. This was discovered late in the development of FASCODE, but was left for future effort. We shall discuss this briefly in the final section.

The computation of the second layer then proceeds using the HIRACC algorithm with the reset value of the sampling interval. The new results are also written to disk, by Subroutine PANEL, using a different file name. Having obtained the spectral absorptance for each layer, one now needs a method to merge the two results such that after the merger one has the absorptance for a path through both layers defined at the resolution of the higher layer. This is obtained by interpolating the "old" or coarser resolution results into the "new" or finer resolution values of the spectral absorptance. A simple four-point Lagrangian interpolation scheme proved adequate[15]. This is summarized as follows: If f(x) is a function defined numerically over a given region with a constant increment h, then the value of the function at some point $(x_0 + ph)$, where p is a real number $(-1 < p < 2)$, is approximated by the relation

$$f(x_0 + ph) = A_{-1}f_{-1} + A_0f_0 + A_1f_1 + A_2f_2 \tag{3.1}$$

Here f_n is the value of the function at $(x_0 + ph)$ and the constants are given by the relations

$$A_{-1} = -p(p - 1)(p - 2)/6 \tag{3.2}$$

$$A_0 = (p^2 - 1)(p - 2)/2 \tag{3.3}$$

$$A_1 = -p(p + 1)(p - 2)/2 \tag{3.4}$$

$$A_2 = p(p^2 - 1)/6 \tag{3.5}$$

The number of allowed resolution ratios was chosen in the following manner. First it was required that the old and new absorptances should be aligned at some running values of the wave number at least. This implied that the ratios were to be ratios of integers. For further simplification it was also required that the ratio be of the form (N + 1)/N. In addition, one must include the case where the sampling interval does not change, namely a one-to-one ratio. After some experimenting with larger values of the integer N, it was found adequate to limit the number of ratios to the following: 2/1, 3/2, 4/3, 5/4 and 1/1.

The interpolation is then performed by identifying the quantity x_o in Equation (3.1) with the next lowest wave number of the old array below the value needed for addition to the new array. The interpolation is characterized by an index (called ITYPE in the program) which is the number of points needing interpolation between the wave numbers which co-align in the two arrays. This will be clear after a glance at Table 3.1. For the 1/1 case (straight add) no interpolation is required and the index is zero. For the 2/1 ratio, one point must be interpolated and ITYPE=1. The remaining schemes are quite clear. The value of the index ITYPE is used to determine the values of p in Equations (3.2) - (3.5) and then in turn to compute arrays of the interpolation constants A_i for interpolating the values between the co-aligned wave numbers. The remainder of the merging algorithm consists in bookkeeping to access the two disk files containing the panel data for the old and the new layers. In addition, the merged results are also written to a third disk file for merging with the next layer. The procedure is

TABLE 3.1. ALLOWED RATIOS AND SCHEMES

Ratio		Scheme	Index (ITYPE)
1/1	Old	● ●	0
	New	● ●	
2/1	Old	● x ●	1
	New	● ● ●	
3/2	Old	● x ● x ●	2
	New	● ● ● ●	
4/3	Old	● x ● x ● x ●	3
	New	● ● ● ● ●	
5/4	Old	● x ● x ● x ● x ●	4
	New	● ● ● ● ● ●	

● = Calculated spectral value

x = Interpolated spectral value to match value in new spectrum then continued until the final layer is processed.

In order to test the merging algorithm, the following procedure was devised. The calculation was performed as described above except that the spectral absorptance calculated for each layer was replaced by a known function of simple form. Consider the case where the absorptances from two layers are merged. If ν_0 is the lower boundary of the frequency range of interest and $\Delta\nu_1$ and $\Delta\nu_2$ are the sampling intervals for layers 1 and 2 respectively, ($\Delta\nu_1 \geq \Delta\nu_2$ by assumption) then the test is made by replacing the calculated optical depths in each layer (e.g., τ_1 and τ_2) by the expressions

$$\tau_1 = \nu_0 + (j_1 - 1)\,\Delta\nu_1/2 \text{ and } \tau_2 = \nu_0 + (j_2 - 1)\Delta\nu_2/2 \tag{3.6}$$

where j_1 and j_2 are running indices which give the number of frequencies at which calculations have been performed, i.e., the lower bound wave number is $j = 1$ and the upper bound is $j = n_\ell$ where n_ℓ is the total number of frequency values in the region calculated.

The array τ_1 is to be merged into the finer spaced array τ_2. Call the merged result τ_{12} and let j be its running index. From Equation (3.6), it follows readily that

$$|\tau_{12}\,(j + 1) - \tau_{12}\,(j)| = \Delta\nu_2 \tag{3.7}$$

provided the merging process is correct. Since the interpolation involved cannot be expected to be exact, the following test was made for all values of τ_2

$$|\tau_{12}\,(j) - \tau_{12}\,(j - 1)| \Big/ \Delta\nu_2 \leq 10^{-4} \tag{3.8}$$

If this inequality was not satisfied the program was directed to print this fact together with the associated parameters.

This testing procedure was followed and the inequality (Equation (3.8)) was found to be satisfied at all points except at the lower and upper bounds of the wave number interval. This was due to the problems of starting and stopping the Lagrangian interpolation when points are not available. A simplified interpolation scheme was chosen at these boundaries which did not have sufficient accuracy to satisfy the merging criterion in Equation (3.8). This is <u>not</u> a problem however, since the HIRACC algorithm[3] automatically expands the requested wave number region by a small amount in order to assure that all lines which contribute to a wave number interval are included. Thus we conclude that the merging procedure does pass our criterion. The extension of this test to more than two layers is straightforward. In Equation (3.6), the factor 1/2 is to be replaced by 1/N where N is the number of layers.

The next test of the algorithm required the calculation of a case for which another high-resolution calculation has been performed. For this purpose, the calculations done by Kyle at NCAR were selected[16]. Kyle performed a multilayer atmospheric transmission calculation in the wave number region (1-2600) cm^{-1}. He used the AFGL tape, a Voigt line shape truncated at 5 cm^{-1} from the line center and a model atmosphere based on the AFGL mid-latitude summer profile[14]. The atmosphere he used is given in Table 3.2. We note briefly that Kyle's results were degraded in resolution by convolution with a triangular instrument scanning function with full width at half maximum of 20, 5, and 0.1 cm^{-1}. Since FASCODE did not as yet have this feature, we compared directly with his highest resolution graphs. It is also noted that Kyle's atmosphere had to be modified to fit the ratio criterion mentioned above. This was done by defining additional layers where there were larger altitude gaps in Kyle's atmosphere. We show the resulting atmospheric model in Table 3.3. We limited ourselves to the wave number range (2000, 2200) cm^{-1}. This shall be referred to as the test problem.

The results of the calculation for our test problem are shown in Figure 4.2. The spectra obtained by FASCODE are in entire agreement with those obtained by Kyle. We delay further discussion of the results until the next section in order to present transmittance and radiance results at the same time.

TABLE 3.2. ATMOSPHERIC LAYERS USED IN THE COMPUTATIONS OF KYLE[16]

Layer	Alt. (km)	Pressure Range (mb)		Temp. (°K)	Number of Molecules in Layer (molecules/cm^2)						
					H_2O	CO_2	O_3	N_2O	CO	CH_4	O_2
1	4	628.	152.	249	1.11×10^{22}	3.32×10^{21}	1.60×10^{18}	2.77×10^{18}	7.55×10^{17}	1.61×10^{19}	2.11×10^{24}
2	14	152.	59.4	216	1.19×10^{19}	6.47×10^{20}	1.90×10^{18}	5.39×10^{17}	1.47×10^{17}	3.14×10^{18}	4.12×10^{23}
3A*		59.4	13.2	223	1.87×10^{19}	3.23×10^{20}	4.20×10^{18}	2.69×10^{17}	7.34×10^{16}	1.56×10^{18}	2.05×10^{23}
3B*	30	13.2	3.32	241	6.40×10^{18}	6.91×10^{19}	1.80×10^{18}	5.76×10^{16}	1.57×10^{16}	3.35×10^{17}	4.40×10^{22}
4	40	3.32	1.75	262	6.40×10^{17}	1.10×10^{19}	2.00×10^{17}	9.15×10^{15}	2.49×10^{15}	5.32×10^{16}	6.98×10^{21}
5	45	1.75	0.58	275	3.70×10^{17}	8.11×10^{18}	6.70×10^{16}	6.76×10^{15}	1.84×10^{15}	3.93×10^{16}	5.16×10^{21}
6	54	0.58	0.00	259	1.00×10^{17}	4.08×10^{18}	8.60×10^{15}	3.39×10^{15}	9.26×10^{14}	1.98×10^{16}	2.59×10^{21}

*The 152-13.2 mb pressure range was too large for a single layer approximation, so an additional layer was added at 59.4 mb. This altitude is not included in the figures.

TABLE 3.3. ATMOSPHERIC LAYERS USED IN THE TEST PROBLEM

Layer	Pressure (mb)	Temp. (K)	Number of Molecules in Layer (molecules/cm^2)						
			H_2O	CO_2	O_3	N_2O	CO	CH_4	O_2
1	525.857	264.	.938E+22	.141E+22	.261E+18	.120E+19	.320E+18	.683E+19	.894E+24
2	332.874	243.	.165E+22	.128E+22	.459E+18	.109E+19	.292E+18	.622E+19	.815E+24
3	197.564	223.	.632E+20	.626E+21	.535E+18	.531E+18	.142E+18	.304E+19	.398E+24
4	124.000	216.	.761E+19	.404E+21	.785E+18	.343E+18	.919E+17	.196E+19	.257E+24
5	77.277	217.	.505E+19	.251E+21	.108E+19	.213E+18	.570E+17	.122E+19	.159E+24
6	36.603	224.	.135E+20	.328E+21	.331E+19	.279E+18	.746E+17	.159E+19	.209E+24
7	8.340	243.	.499E+19	.705E+20	.126E+19	.598E+17	.160E+17	.342E+18	.448E+23
8	2.551	263.	.491E+18	.112E+20	.153E+18	.948E+16	.254E+16	.542E+17	.710E+22
9	1.156	267.	.259E+18	.844E+19	.660E+17	.716E+16	.192E+16	.409E+17	.536E+22
10	.274	255.	.475E+17	.408E+19	.118E+17	.346E+16	.928E+15	.198E+17	.259E+22

4.0 RADIANCE FROM AN ATMOSPHERE IN LOCAL THERMODYNAMIC EQUILIBRIUM

If one assumes that a given infinitesimal volume of a gas is in local thermodynamic equilibrium (LTE) at some temperature, it follows that whatever amount of radiant energy is absorbed by this gas, an equal amount of energy must be re-emitted in order to maintain the equilibrium state. Furthermore, the spectrum of the radiation re-emitted will be determined by the black body or Planck function using Kirchhoff's law. This LTE model may be expected to be more nearly valid at the lower altitudes where collision frequencies are quite high. At the higher altitudes ($\gtrsim$ 25 km), one should expect non-equilibrium effects (NLTE) to be important. Indeed, Degges[17] has been developing a comprehensive NLTE atmospheric radiance model for a number of years and other workers have studied the high altitude problem as well[18]. The FASCODE Program could also be used for the NLTE case if vibrational or rotational temperatures and/or populations of states at higher altitudes were read in. Such data could be prepared by programs such as Degges' work.

Despite its shortcomings at higher altitudes, a LTE atmospheric radiance model can be very useful not only in the low altitude regime, where it rests on solid footing, but also in the higher regions where one may use it to characterize the degree of deviation of the NTLE situation from the equilibrium case. A number of workers have prepared LTE radiance models among which we want to mention the recent extension of the AFGL LOWTRAN model to include a radiance calculation[2].

In this section we present the algorithms implemented in FASCODE to enable the calculation of atmospheric radiance along a given optical path assuming LTE along that path. To provide the most efficient calculation of the radiance, it was decided to calculate its value layer-by-layer along with the spectral absorptance calculation described in the previous section. It should be noted that the radiance of a path proceeding from lower toward higher levels is not the same as that for the same geometrical path proceeding from higher to the lower layers. Figure 4.1 presents a sketch of two such paths, space-to-ground (Figure 4.1a) and ground-to-space (Figure 4.1b). A four-layer atmosphere has been shown for simplicity, each layer being labeled by the letter A, B, C, or D. The boundaries between each layer have

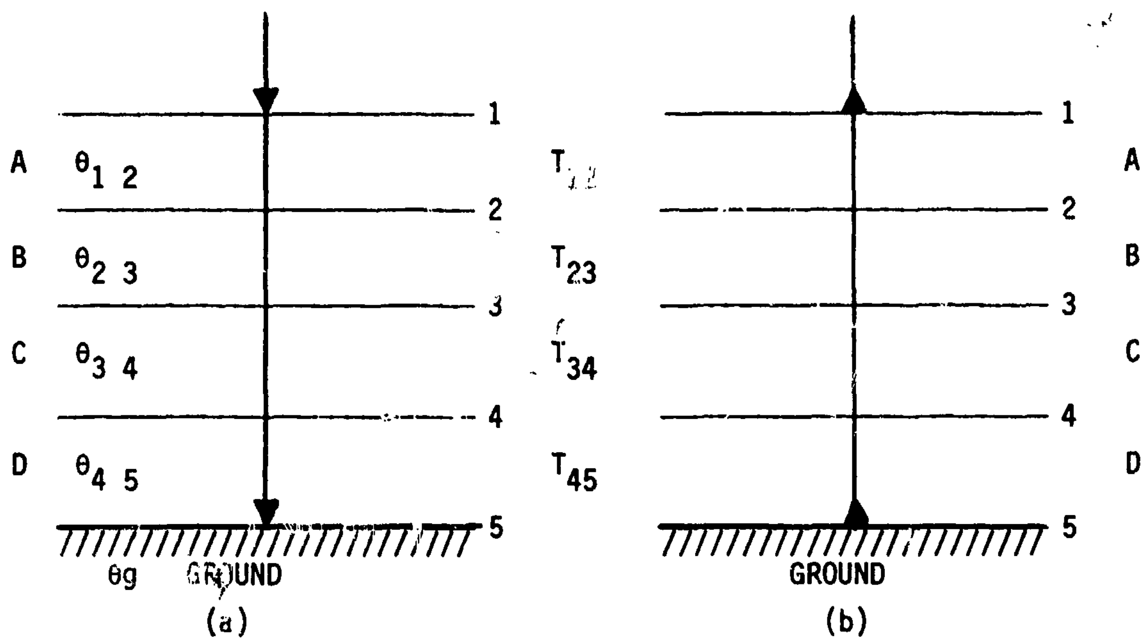

Figure 4.1: Sketch of the difference between a path looking up and one looking down.

been numbered from 1 to 5 where 5 is the ground at temperature at θ_g. The temperature of each layer is called $\theta_{i,\ i+1}$ where i is one of the labels for the boundaries. The transmittance contributions between any two boundaries are defined as T_{ij} where i and j are boundary indices.

Consider now the equilibrium radiance from Layer A to space in Figure 4.1a. Using Kirchhoff's law this may be written as

$$R_{A-S} = (1 - T_{12})\ P(\theta_{12}, \nu) \tag{4.1}$$

where P is the Planck function and ν is the frequency in cm^{-1}. Proceeding to the next layer the contribution of this layer to the radiance observed in space is

$$R_{B-S} = T_{12}(1 - T_{23})\ P(\theta_{23}, \nu) \tag{4.2}$$

and the remaining two layers can be written as

$$\begin{aligned} R_{C-S} &= T_{12}\ T_{23}\ (1 - T_{34})\ P(\theta_{34}, \nu) \\ &= T_{13}(1 - T_{45})\ P(\theta_{34}, \nu) \end{aligned} \tag{4.3}$$

and

$$\begin{aligned} R_{D-S} &= T_{12}\ T_{23}\ T_{34}(1 - T_{45})\ P(\theta_{45}, \nu) \\ &= T_{14}(1 - T_{45})\ P(\theta_{45}, \nu) \end{aligned} \tag{4.4}$$

Finally the contribution of the ground is

$$R_{g-s} = T_{12}\ T_{23}\ T_{34}\ T_{45}\ P(\theta_g, \nu) = T_{15}\ P(\theta_g, \nu) \tag{4.5}$$

The total radiance is then the sum of all these terms. Extending this to the case of n layers with n + 1 boundaries we can write

$$R_{\text{total}}^{(\text{down})} = \sum_{i=1}^{n} T_{1,i} \, (1 - T_{i,\, i+1}) \, P(\theta_{i,\, i+1}, \nu) \tag{4.6}$$

$$+ T_{1,\, n+1} \, P(\theta_{n+1}, \nu)$$

where now θ_{n+1} is a temperature characteristic of the last boundary. Note that in general, this is not necessarily the ground. (It shall be noted $T_{ij} = \prod_{k=1}^{j} T_{k,\, k+1}$.) Also, it is not required that the spectral distribution at the boundary be given by the Planck function. An arbitrary spectrum or one characteristic of a particular type of radiating boundary may be substituted for the (n+1)'st term.

Turning now to the other case, looking up, we can write the following expression for the contribution to the radiance from each of the layers

$$R_{5-4} = (1 - T_{45}) \, P(\theta_{45}, \nu) \tag{4.7}$$

$$R_{4-3} = T_{45}(1 - T_{34}) \, P(\theta_{34}, \nu) \tag{4.8}$$

$$\begin{aligned} R_{3-2} &= T_{34} \, T_{45} \, (1 - T_{23}) \, P(\theta_{23}, \nu) \\ &= T_{35} \, (1 - T_{23}) \, P(\theta_{23}, \nu) \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} R_{2-1} &= T_{23} \, T_{34} \, T_{45} \, (1 - T_{12}) \, P(\theta_{12}, \nu) \\ &= T_{25} \, (1 - T_{12}) \, P(\theta_{12}, \nu) \end{aligned} \tag{4.10}$$

Again, the total radiance is the sum of the individual terms for each layer which may be written as

$$R^{(up)}_{(total)} = \sum_{i=0}^{n-1} T_{n-i,\ n} \left(1 - T_{n-i-1,\ n-i}\right) P\left(\theta_{n-i-1,\ n-i},\ \nu\right)$$

$$+ T_{1,\ n+1}\ P(\theta_1,\ \nu) \tag{4.11}$$

where we have implicitly defined $T_{1,1} = 1$ for convenience. To present a radiating boundary at the end of an upward looking path, such as a cloud or the Zodiacal light, one may add the same type of boundary radiating term as discussed for the downlooking case. An upward looking boundary has not been included in FASCODE as yet but a user can readily add one if it should be required. A glance at Equations (4.6) and (4.11) is sufficient to see the difference between each case.

The expressions in (4.6) and (4.11) are convenient mathematical representations of the algorithm but for computational purposes it is more useful to represent the algorithms such that they manipulate the new increment to the transmittance for a given layer i, and the radiance and total transmittance accumulated up to that layer. As was mentioned in Section 3, the calculation proceeds from the lowest layer to the highest in order to minimize the time for merging. If ΔT_i is the incremental transmittance, E(i) the radiance and P(i) the Planck function, the radiance and transmittance after the i^{th} layer has been traversed are

$$\overset{(up)}{E(i)} = \overset{(up)}{E(i-1)} + (1 - \Delta T_i)\ P(i) \cdot T(i-1) \tag{4.12}$$

and

$$\overset{(down)}{E(i)} = \Delta T_i\ \overset{(down)}{E(i-1)} + (1 - \Delta T_i)\ P(i) \tag{4.13}$$

for the radiance looking up and looking down, respectively. The transmittance is clearly given by the relation (for both cases)

$$T(i) = \Delta T_i \cdot T(i-1) \tag{4.14}$$

The boundary radiance is frequently not a function with very fine spectral resolution. Thus, for the lookingdown case, the most efficient procedure is to compute the boundary radiance at the coarse resolution of the lowest layer and include it with the radiance of the first layer (modified by the transmittance, of course). This will then be properly carried along through the remainder of the calculation. For the case of a radiating boundary at the upper end of a path looking toward space, it is simplest to add the contribution from the boundary after all calculations have been performed modifying it by the total transmittance computed for the path. As mentioned above, FASCODE at this juncture does not include coding for this second case, but does include a radiating boundary at the end of a down-looking path.

In order to speed up the calculation of the Planck function the following procedure was followed. For each panel, the black body function is computed and the results stored in an array with separation of one wave number. The value needed at a given wave number is obtained by interpolation which is performed only when the value of the Planck function can be expected to have changed sufficiently to warrant an updated value. If the Planck function is written

$$P(\nu) = A\nu^3 \left(e^{s\nu} - 1\right)^{-1} \tag{4.15}$$

where ν is in wave numbers and $s = C_2/\theta$, θ being the temperature, and C_2 the second radiation constant (1.4388). Taking the derivative with respect to ν we may write

$$\frac{dP}{P} = \frac{d\nu}{\nu}\left[3 - (s\nu)\left(1 - e^{-s\nu}\right)^{-1}\right] \tag{4.16}$$

If one takes $|\Delta P/P| \lesssim 10^{-4}$, the increment at which one should interpolate to obtain a new value of the Planck function is easily computed. This alleviates the need to recalculate the Planck function unecessarily.

The implementation of this algorithm for LTE radiance in FASCODE is contained in two subroutines, EMUP and EMDOWN, for paths "looking" up toward space and "looking" down toward the ground respectively. An additional subroutine was prepared to compute the first layer (the lowest), EMINIT. In this routine the possibility of radiation from a boundary was included for the downward-looking case.

The merging of the results layer-by-layer proceeds in a similar fashion to that for the absorptance described in Section 3. One difference is that now one needs to merge two quantities, namely, the radiance computed and the transmittance as the calculation proceeds from layer to layer. These quantities are written to disk in two records, first the radiance, followed by the transmittance. The code accesses the HIRAC algorithm to obtain the spectral absorptance panel by panel for each layer. This information is converted to transmittance by exponentiation in either EMUP or EMDOWN depending on the case of interest. The LTE radiance algorithm is then exercised as the merging is taking place.

The radiance routines were tested in a manner similar to the merging test described in the last section. Consider the case of a series of n layers with a path from space to ground with a boundary at temperature θ. If one modifies the input data such that each layer has the same temperature as the boundary, then one can show that the resulting radiance will be given by the Planck function at temperature θ, a result which is not at all surprising. The same result holds for the upward looking case if a boundary is added at the upper end and the temperatures are all set equal. This procedure was followed with the results as expected to within accuracy requirements inherent in the method.

The radiance package has been tested for the problem described in the previous section. The radiance results are shown in Figures 4.2 together with the transmittance for a path looking from space to ground which is modeled as a black body radiator with temperature, T = 273 K. Note the smaller range of the abscissa for Figures 4.2(e) and 4.2(f). The absorption features seen in Figures 4.2 are clearly seen in the radiance profiles as well. The expanded frequency scale in Figures 4.2(e) and 4.2(f) show this most clearly. Self-reversal can easily be seen in the radiance for strong absorption lines.

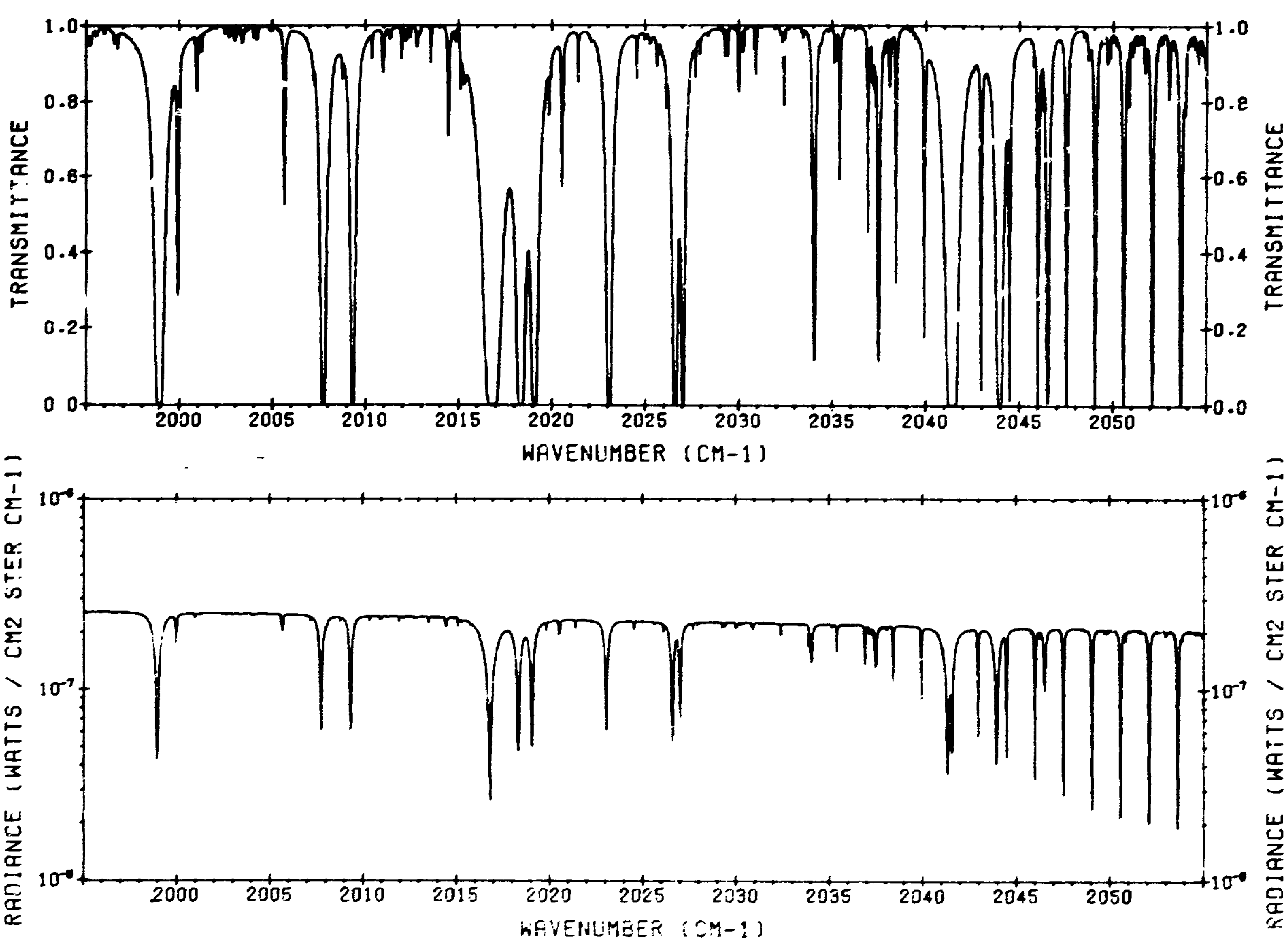

Figure 4.2 (a): Transmittance and radiance for the test problem (1995-2055 cm^{-1})

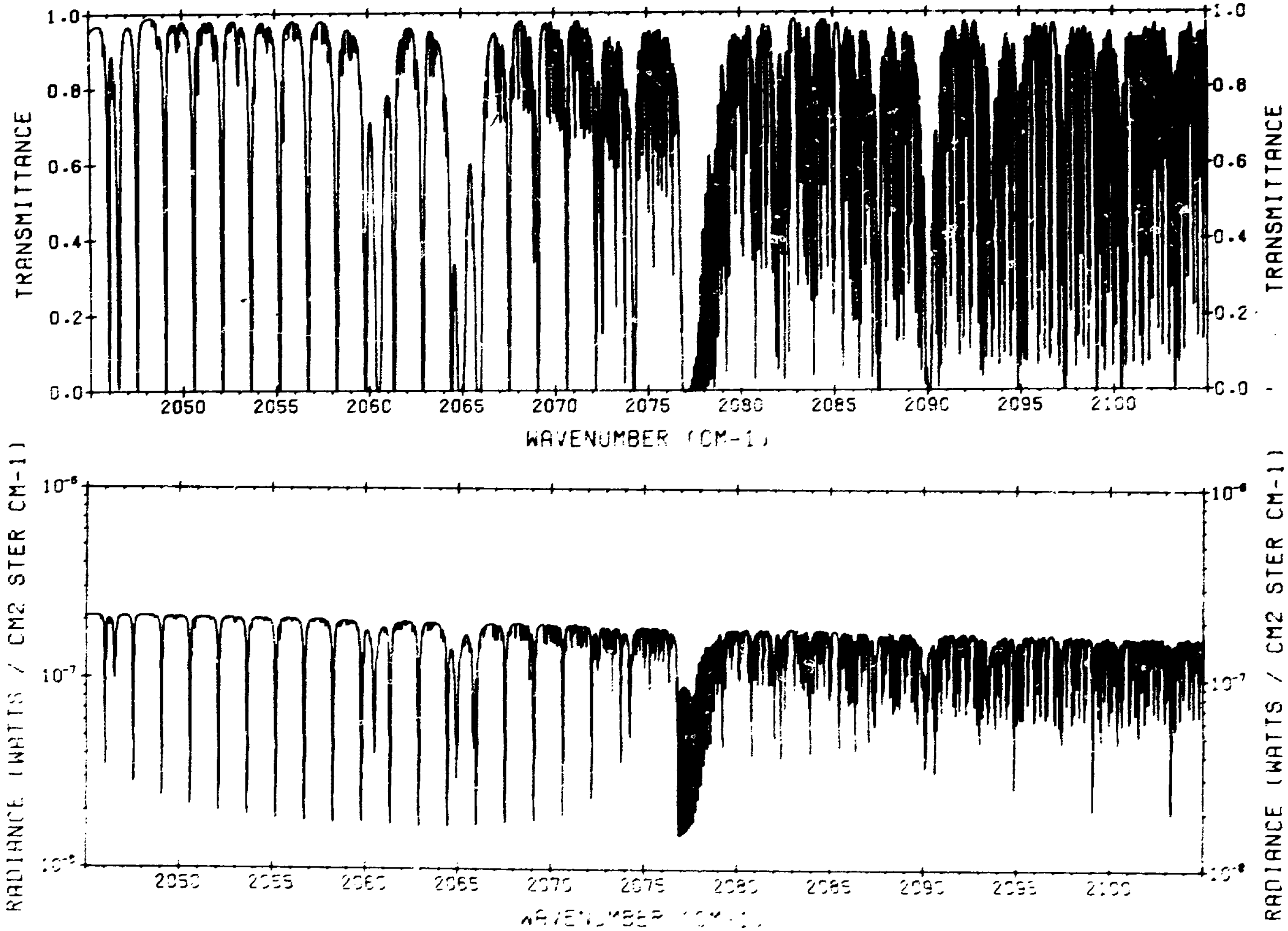

Figure 4.2 (b): Transmittance and radiance for the test problem (2045-2105 cm^{-1}).

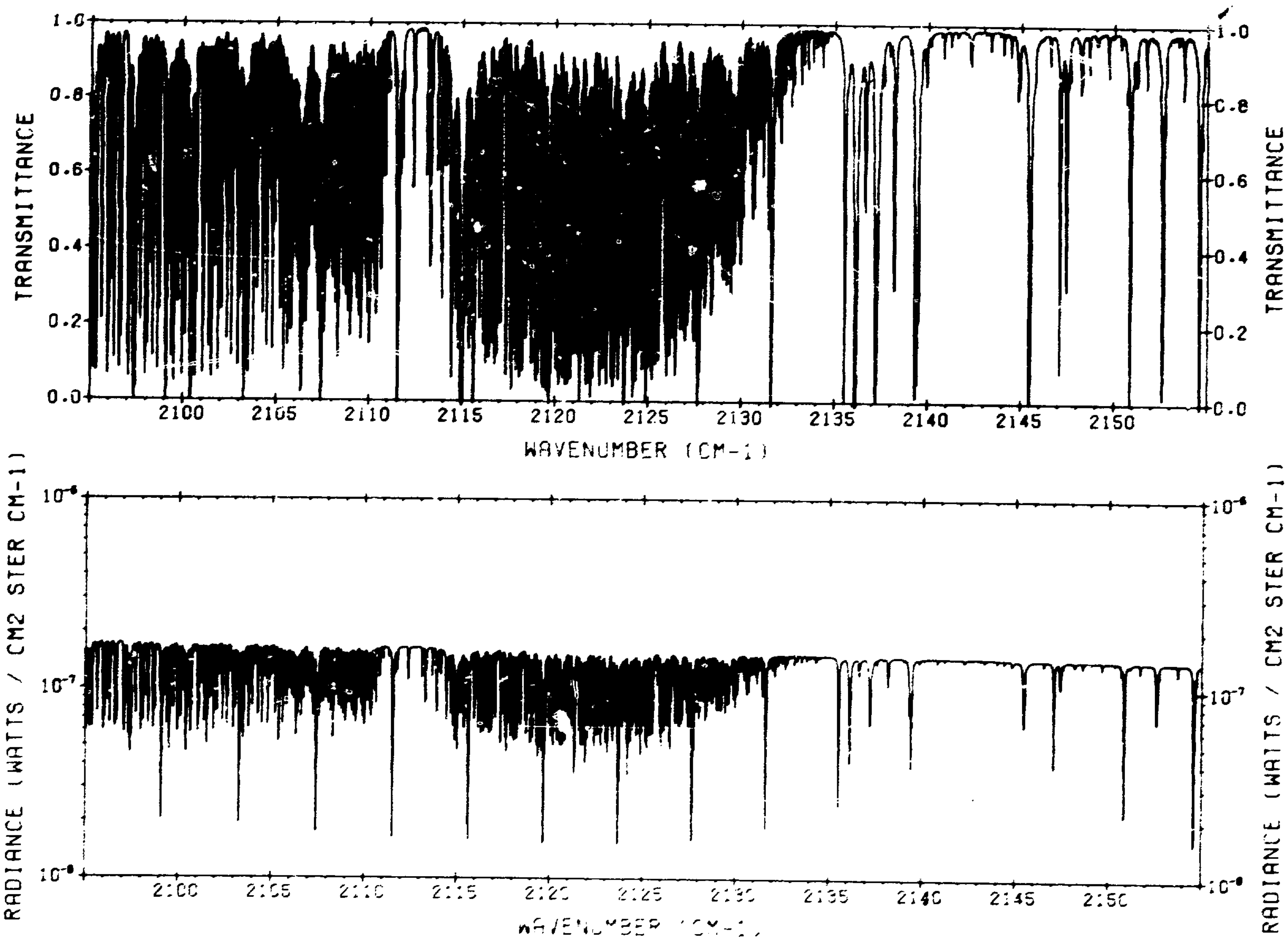

Figure 4.2 (c): Transmittance and radiance for the test problem (2095-2155 cm^{-1}).

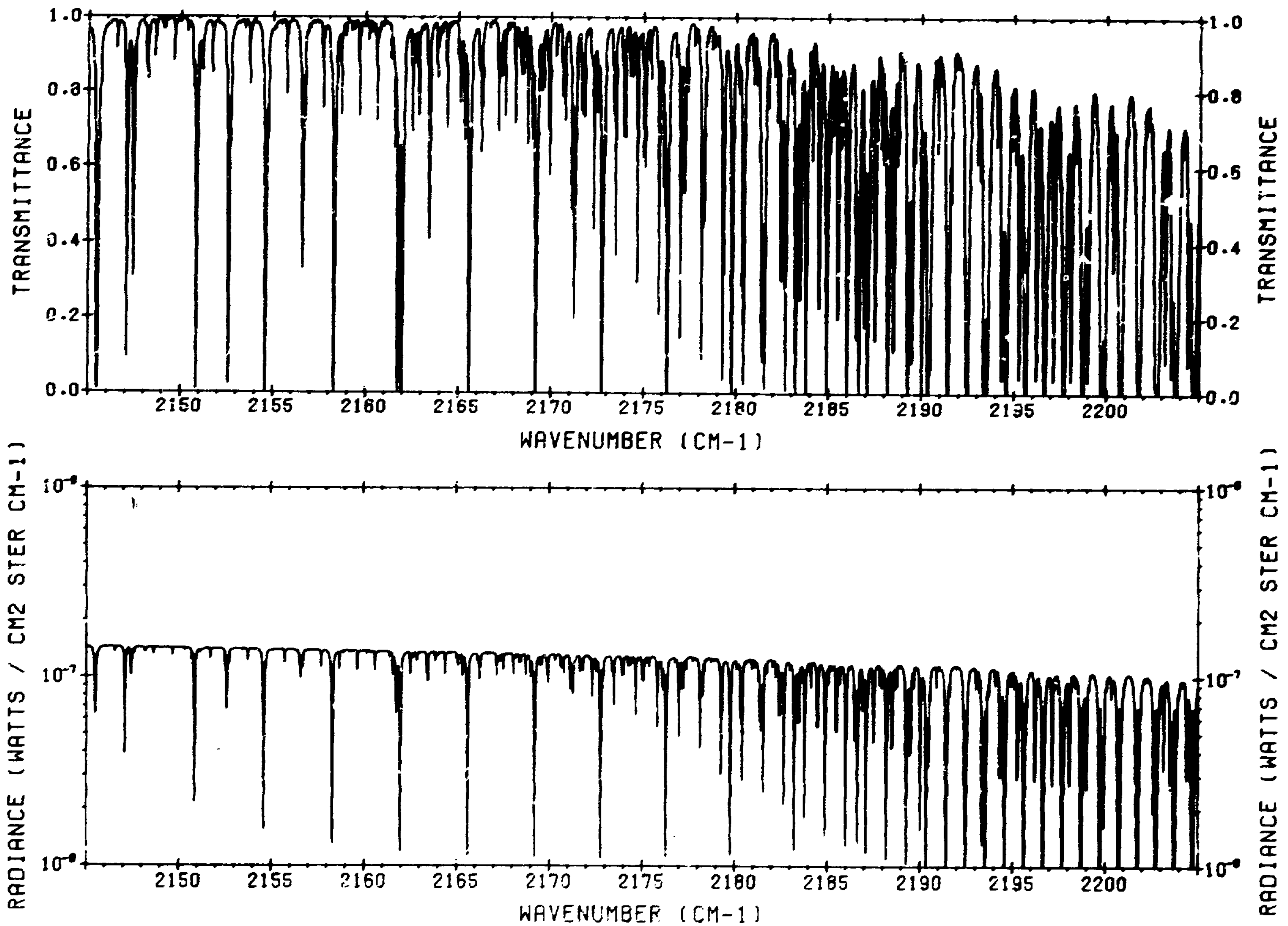

Figure 4.2 (d): Transmittance and radiance for the test problem (2145-2205 cm^{-1}).

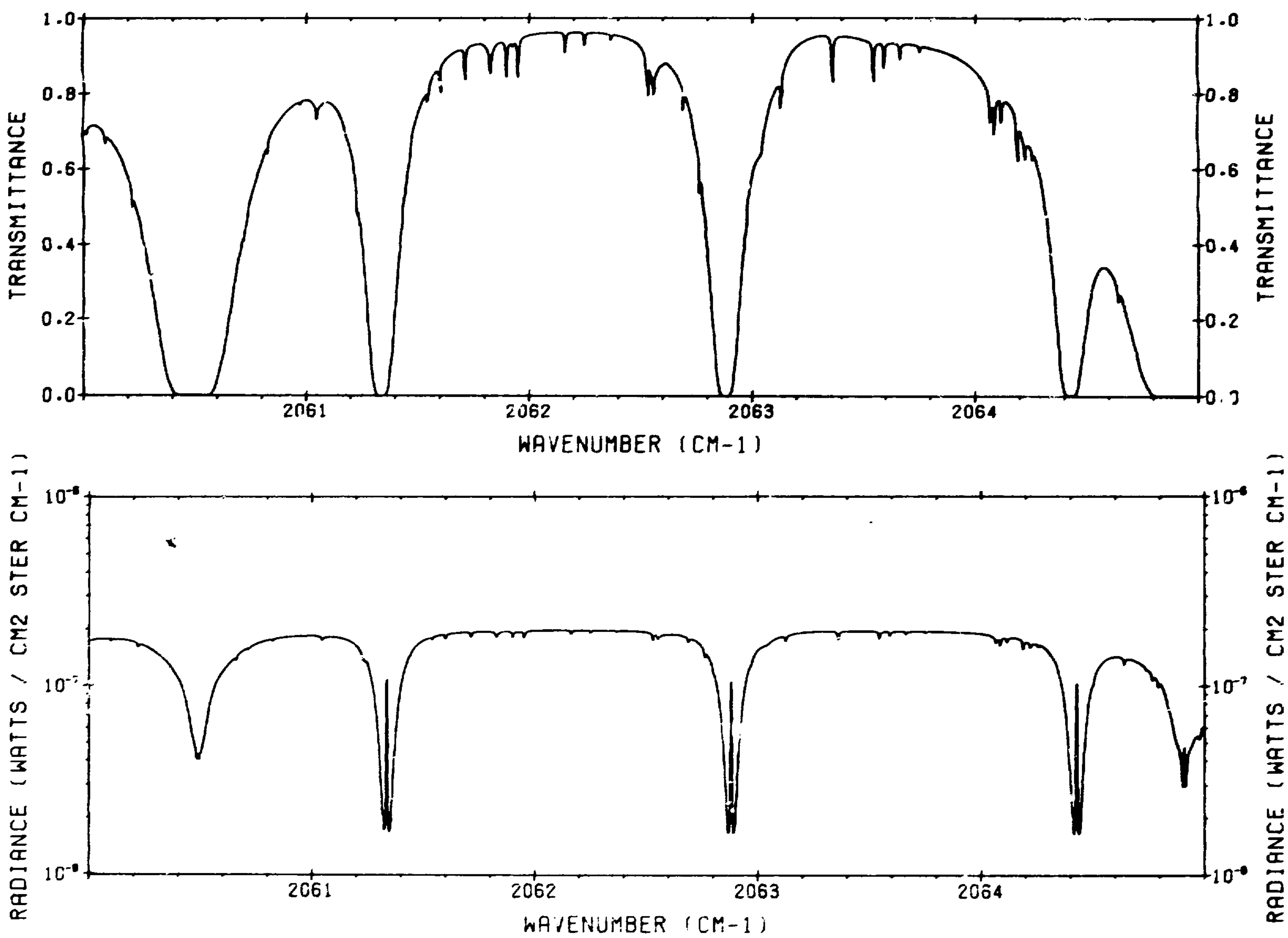

Figure 4.2 (e): Transmittance and radiance for the test problem with expanded scale (2060-2065 cm^{-1}).

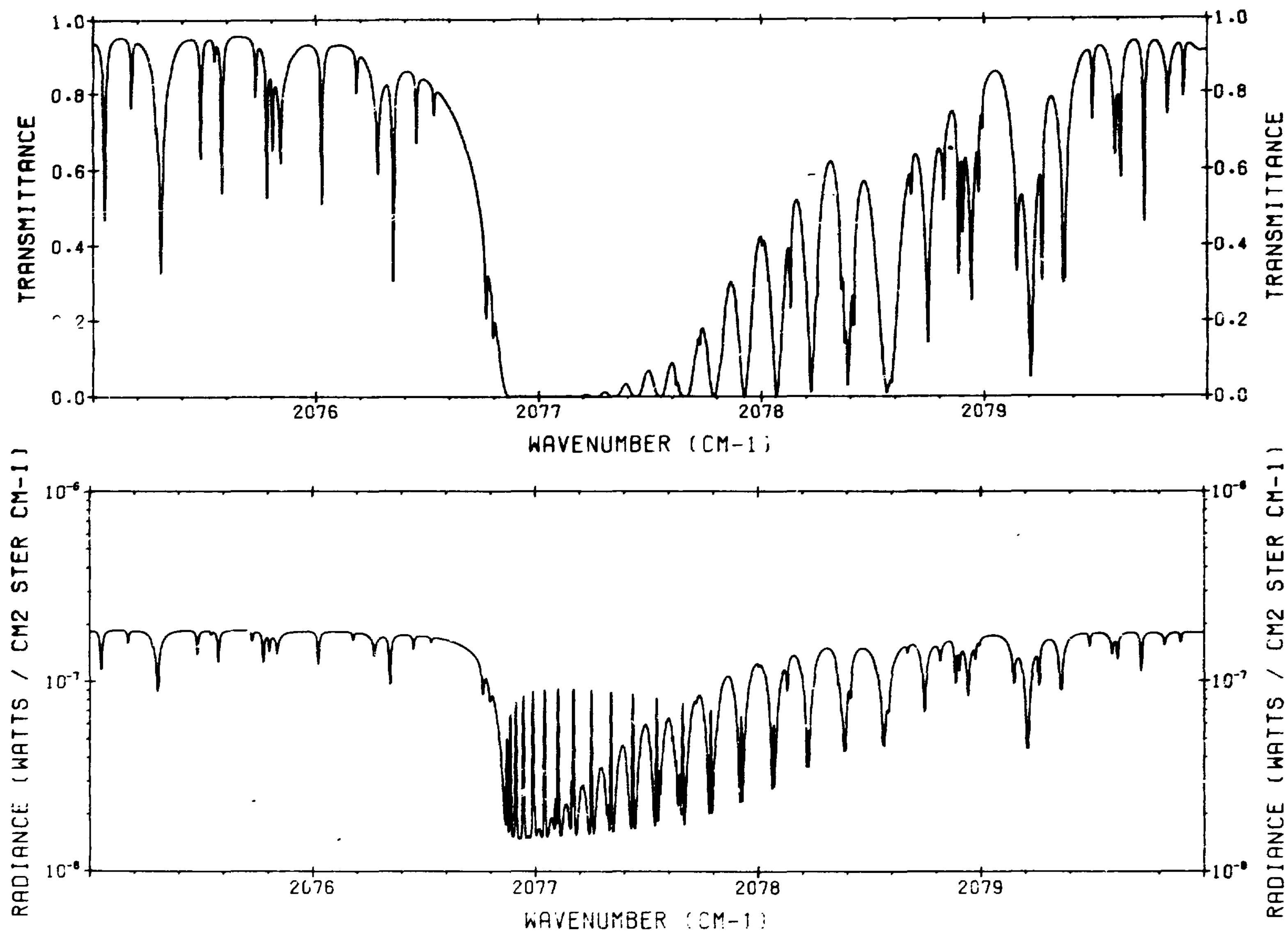

Figure 4.2 (f): Transmittance and radiance for the test problem with an expanded scale (2075-2080 cm^{-1}).

This is particularly dramatic in Figure 4.2(f) near the band edge of the CO_2 (11101-00001) transition.

In Table 4.1 we give the time results of the run made to produce Figures 4.2. All times are for the CDC 6600 computer system at AFGL. For each of the layers of the test problem, we give the altitude boundaries and the temperature and pressure characterizing each layer. The next column gives the value of ζ used in computing the Voigt shapes for that layer. The average linewidth and the sampling interval are given in the columns labeled ALPHA and DV respectively. The ratios of the sampling intervals are given in the following column. The final four columns show the timing for the convolution of the spectra (CONV), the writing of the panels to disk (PANEL), the calculation of the radiance (EM) and the calculation of the transmittance merging (ABS). The sums of the times for each column are also presented. Note that the total number of lines processed was 6681. The convolution took approximately 0.7 msec per line per layer. This latter statistic is very meaningful, since it gives the reader some idea of the speed of the program. The total time for a transmittance calculation only, may be obtained by summing the totals of the columns marked CONV, PANEL and ABS (i.e., 102.2 seconds). For the radiance calculation along the same path, one obtains the total time by summing the totals of the columns marked CONV, PANEL and EM (i.e., 175.3 seconds).

TABLE 4.1. TIMING RESULTS FOR THE TEST PROBLEM

Layer	Alt. km	Press. mb	Temp. K	Zeta	Alpha cm^{-1}	DV cm^{-1}	$\frac{DV_i}{DV_{i-1}}$	CONV sec	PANEL sec	EM sec	ABS sec
1	4-7	525.9	264.4	0.95	0.04396	0.01380		4.6	0.2	0.7	
2	7-11	332.9	242.7	0.94	0.02919	0.00920	3:2	4.5	0.3	1.8	0.8
3	11-14	197.6	222.9	0.90	0.01823	0.00613	3:2	4.1	0.4	2.7	1.1
4	14-17	124.0	216.0	0.86	0.01181	0.00307	2:1	5.2	0.9	5.1	2.2
5	17-20	77.27	216.9	0.79	0.00763	0.00230	4:3	4.5	1.2	7.1	3.1
6	20-30	36.60	223.9	0.64	0.00419	0.00115	2:1	4.9	2.4	13.5	5.8
7	30-40	8.340	243.0	0.28	0.00230	0.000575	2:1	5.3	4.7	26.6	11.7
8	40-45	2.551	263.3	0.10	0.00207	0.000575	1:1	4.8	4.7	15.4	2.0
9	45-54	1.156	267.3	0.048	0.00202	0.000575	1:1	4.7	4.7	15.5	2.1
10	54-∞	0.2742	255.4	0.012	0.00193	0.000575	1:1	4.5	4.7	15.6	2.1
						TOTAL		47.1	24.2	104.0	30.9

5,681 Lines 2000 to 2200 cm^{-1}

5.0 CONCLUSIONS AND RECOMMENDED EXTENSIONS OF THE PROGRAM

For certain purposes it is helpful to use a pure Lorentz or a pure Doppler line profile instead of the Voigt profile, which takes slightly more computational time than the Lorentz line shape or the Doppler line shape alone. For example, a user might have a problem involving only transmission at the very high layers of the atmosphere, ($\gtrsim$ 40 km) in which region the line shape is purely Doppler. On the other hand, one might be interested in studying a laboratory experiment at relatively high pressures for which the Lorentz profile is adequate. The addition of the pure Doppler and the pure Lorentz cases is straightforward and has been accomplished. The reader is referred to the program listing in Appendix 3. The revisions which were required for the pure Lorentz case include:

a. Subroutine SHAPED is not needed.

b. The least squares fits for the linear combination of the two profiles are not necessary (AVRAT, $a_{VD}(\zeta)$, $a_{VL}(\zeta)$, $C(\zeta)$).

c. All program references to the quantity ζ are deleted.

d. The final result is, of course, similar to the original HIRACC coding[3]. Here it is called HIRACL.

For the pure Doppler case, the following differences arise:

a. Subroutine SHAPEL is not used.

b. Delete all ζ references.

c. Only the FF array is used, VSF and SF are not needed.

d. The resulting routine is called HIRACD.

Some care had to be taken to assure that the proper indexing is made.

The contributions of the various continuum features have not been included as yet[1,2,14,20]. These features arise from a number of physical processes such as absorption by atmospheric aerosols, and a variety of molecular continuum processes for molecules such as N_2, CO_2, H_2O. In addition the contributions of the lines beyond 64 half-widths must also be included as part of the continuum contribution. In general, the continuum absorption is a slow function of wave number and may be directly incorporated into the VSF array. This operation will not significantly affect the running time of the program.

The line-by-line results have finer spectral detail than is required for comparison with some experiments. Some instrumentation (especially that used for engineering systems) does not have resolution such that the final spectral detail computed in this version of FASCODE can be resolved. A method is required for convolving a given instrument scanning function with the FASCODE output in order to degrade the detailed results for comparison with lower resolution data. Work on this aspect is currently underway at AFGL. We note that the convolution techniques used in the line-by-line spectral synthesis (HIRACC algorithm) may also be applied to a scanning function convolution.

When the sampling interval is constant from one layer to the next, and one is not calculating the radiance, it seems clearly possible to devise a new method which would decrease the running time considerably. This method would compute at the same time the spectrum for all of those adjacent layers which have a constant sampling interval. For example, above a certain altitude where the Doppler line shape becomes dominant, essentially all of the sampling intervals can be taken to be constant and the merging calculations for all of these layers can be done panel by panel.

Finally, for applications in systems studies it is recommended that spherical geometry be added which could account for the fact that the earth's atmosphere is not plane parallel, but rather spherical. With this addition to the code, the calculation of limb radiance would be easily performed and practical applications could be taken directly from FASCODE.

To illustrate this, consider a remote sensing satellite viewing the earth on some sight path. If the sight path ends on the earth's surface or some other surface above the earth such as a cloud layer, no great extension to the program need be made. One merely needs to program a method for computing column densities of the absorbing molecules for each layer. However, for the case of a sight path that is glancing, or tangent to some altitude above the earth, the problem is more complicated.

Figure 5.1 is a sketch of this case. Here r_{max} is the earth-centered radius to the maximum altitude for which calculations are to be made, and r_T is the corresponding radius to the tangent height at Point T. Let A and B be the points at which the line-of-sight intersects the circle of radius r_{max}. First, we consider the case where only the transmittance is to be calculated. It is clear that the largest sampling interval which will be determined from the criteria programmed in FASCODE, will be found in the tangent height layer and the smallest will be for the layers ending at the two points A and B. Thus the optimum procedure will be to start at the tangent height where one has the smallest number of points needed to characterize the convolved spectral transmittance. Next, we note that the two paths TA and TB are identical compositions of column densities. Thus the transmittance from path TB is identical to that on path TA and the total transmittance may be obtained simply by doubling the contributions from each layer and computing only one of the two paths TA or TB. Note that it is the column densities which are to be doubled and <u>not</u> the transmittance. Also we have tacitly assumed that the atmospheric composition profiles do not change appreciably along the path ATB. Since the angle ACB can be as large as ~ 20°, this may not be the case. Thus, for example, the sight path may enter the atmosphere at ~ 45°N latitude and exit at 65°N latitude and the profiles can be quite different, especially for water. This is a complication which may or not be important in a given case. No essential difficulties should occur, however, should this situation need to be investigated.

For the radiance calculation, it is again best to start at the tangent part of the sight path. This can be done simultaneously with the TB part corresponding to a path looking up and the TA part to a down-looking path. When the calculations for the two final layers are finished, the total radiance, R_{TOT}, is computed simply by the relation

$$R_{TOT} = R_{TA} + T_{TA} R_{TB} \tag{5.1}$$

where R_{TA} and R_{TB} are the results of the radiance calculations on paths TA and TB respectively, and T_{TA} is the transmittance from path TA. If Point B

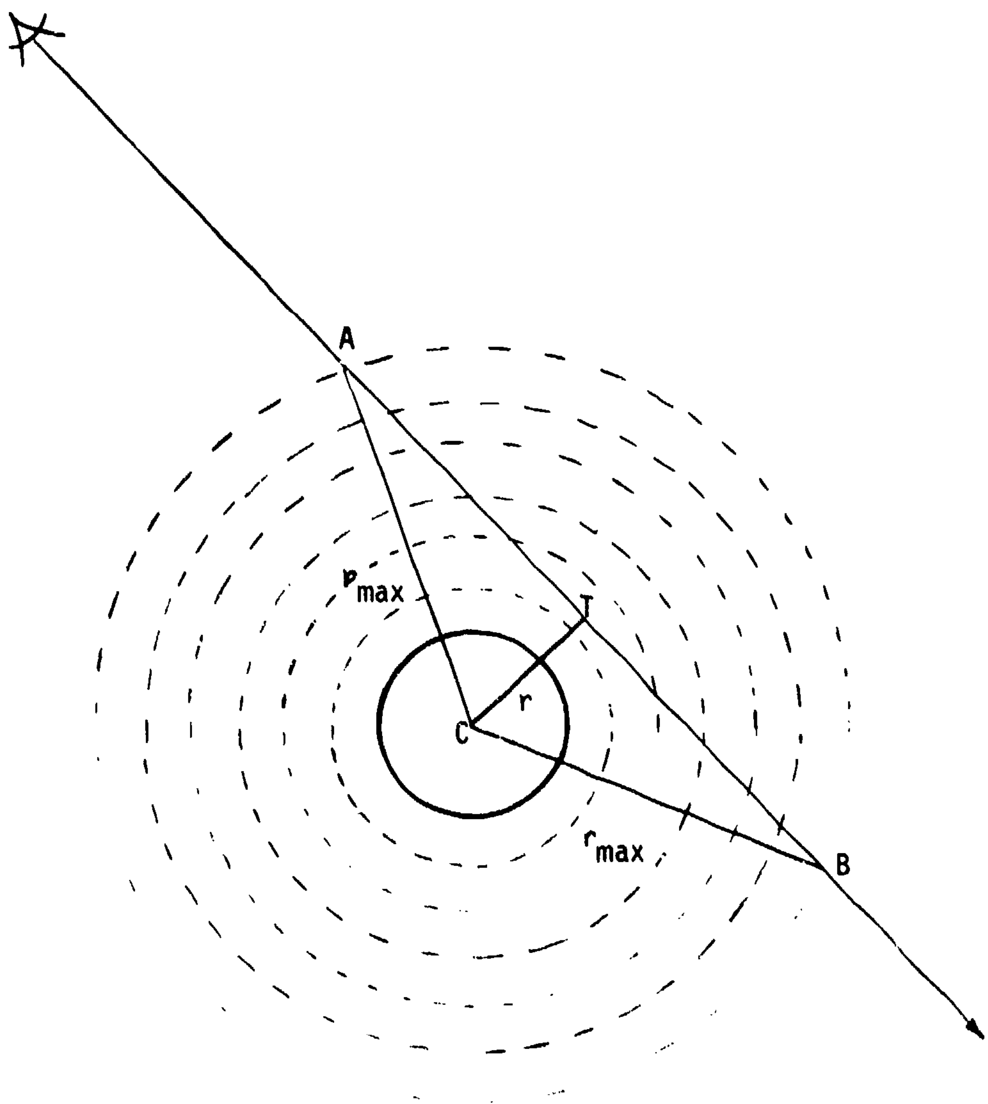

Figure 5.1: Sketch of Tangent Path Satellite Viewing Geometry.

is not at the same altitude as Point A (for example Point B corresponds to a target or a cloud), no essential problems occur. One merely stops the calculation along TA at the altitude of Point B, performs the composition of Equation (5.1) and continues the calculation to Point A as before, but using the value R_{TOT} for the radiance at the stopping point. This method has not been implemented as yet in FASCODE but the implementation should be straightforward.

REFERENCES

1. McClatchey, R.A. et al "AFCRL Atmospheric Absorption Line Parameters Compilation", AFCRL-TR-73-0096 (1973).

2. Selby, J.E.A., Kneizys, F.X. Chetwynd, J.H. and McClatchey, R.A., "Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 4", AFGL-TR-78-0053, 28 February 1978.

3. Clough, S.A., Kneizys, F.X., Chetwynd, Jr., J.H., "Algorithm for the Calculation of Absorption Coefficient-Pressure Broadened Molecular Transitions", AFGL-TR-77-0164, 22 July 1977.

4. Burch, D.E., Gryvnak, D.A., Patty, R.R., and Bartky, C.E., "The Shapes of Collision Broadened CO_2 Lines", Philco Ford Corp., Aeronutronic Report U-3203.

5. Winters, B.H., Silverman, S. and Benedict, W.S., "Line Shape in the Wing Beyond the Band Head of the 4.3 μm Band of CO_2", J. Quant. Spect. Rad. Trans. Vol. 4, p. 527 (1964).

6. Anderson, P.W., Physical Review, Vol. 76, p. 647 (1949).

7. Tsao, C.J., Curnutte, B., J. Quant. Spect. Rad. Trans. Vol. 4, p. 453 (1964).

8. Benedict, W.S. and Kaplan, L.D., J. Chem. Phys., 30, p. 388 (1959).

9. Pierluissi, J.H., Vanderwood, P.C. and Gomez, R.B., J. Quant. Spect. Rad. Trans., Vol. 18, p. 555 (1977).

10. Drayson, S.R., J. Quant. Spect. Rad. Trans., Vol. 16, p. 611 (1976).

11. Armstrong, B.H., J. Quant. Spect. Rad. Trans., Vol. 7, p. 61 (1967).

12. Whiting., E.E., J. Quant. Spect. Rad. Trans., Vol. 8, p. 1379 (1968).

13. Kielkopf, J.F., J. Opt. Soc. Am., Vol. 63, p. 987 (1973).

14. McClatchey, R.A. et al, "Optical Properties of the Atmospnere (Third Edition)", AFCRL-72-0497, Environmental Research Papers, No. 411, 24 August 1972.

15. Abramowitz, M. and I.A. Stegun, "Handbook of Mathematical Functions" NBS Applied Mathematics Series (55), 1964.

16. Kyle, T.G. and A. Goldman, "Atlas of Computed Infrared Atmospheric Absorption Spectra", NCAR-TN/STR-112, November 1975.

17. Degges, T.C., "A High Altitude Radiance Model", AFGL-TR-77-0271, December 1977 (In Press).

18. See for example, J.B. Kumer, "Atmospheric CO_2 and N_2 Vibrational Temperature at 40 to 140 km Altitude", J.G.R., 82, 2195-2202.

19. Burch, D.E., "Semi-Annual Technical Report: Investigation of the Absorption of Infrared Radiation by Atmospheric Gases", Aeronutronic Report U-4784 (31 January 1970).

20. Burch, D.E., Gryvnak, D.A., Patty, R.R., and Bartky, C.E., "Absorption of Infrared Radiant Energy by CO_2 and H_2O. IV. Shapes of Collision-Broadened CO_2 Lines", J. Opt. Soc. Am., Vol. 59, p. 267 (1969).

21. Herzberg, G., Infrared and Raman Spectra, Van Nostrand, New York, (1945).

22. Varanasi, P. and Ho, F.K., "Intensity and Transmission Measurements in the ν_3 Fundamental of N_2O at Low Temperatures", Thirty-Second Symposium on Molecular Spectroscopy, Paper RF5, Columbus, Ohio (1977).

APPENDIX A

CODE STRUCTURE AND DESCRIPTION

In this appendix we present a description of the code. For the reader's convenience much of the documentation of the HIRACC algorithm[3] has been reproduced here with an occasional modification. In addition, the new subroutines developed in this effort are described.

The program has been written to use line parameter input data consistent with that contained in the AFGL line parameter tape[1]. The line data has been reformatted onto a binary file which contains the line data pertinent to the molecules and wave number range of interest. This step has been taken to keep read time consistent with the time required to perform the calculational part of the program. The control parameters are read from the input file and written to the output file; TAPE3 is the binary file containing the line parameter data; and TAPE12 is always the binary output file. The final output and intermediate output files contain a header record which includes the identification information, SECANT, temperature, pressure, molecular identification, and molecular column densities of the homogeneous layers. The first record for each output panel is a header record for the panel which contains the wave number values of the first and last absorption coefficient values of the panel, the wave number increment between output points and the number of output points. The second record of the panel contains either the array of absorption coefficient values resulting from the convolution when only absorptance is calculated or the radiance array followed immediately by the transmittance array for radiance calculations. The current version of the program outputs a maximum of 2400 values per panel; in general, the first and last panel are shorter. The output file MFILE, and an additional file, LFILE, are used to store the intermediate data prior to merging the results for the layers. The merged results are always on MFILE. The file KFILE contains the absorption coefficients for each layer. Table A1 outlines the use of these files and file labels.

FASCODE has been designed to be used with the CDC segmentation feature which allows the program to be run using a minimum of central memory by loading dynamically only those subprograms which are in use at a given stage of the code's execution. Using this feature, FASCODE requires only $\sim 56\ K_8$ words of central memory. If a user does not have a segmentation

TABLE A1. FILES USAGE

<u>UNIT NAMES:</u>

INPUT(=TAPE5), OUTPUT(=TAPE6)	-	Standard I/O files
TAPE3	-	Absorption line data file in buffered binary format
TAPE10	-	Output file from HIRACV routine; labeled KFILE; buffered binary format. Data to be merged with data from previous layer (on LFILE) except for first layer.
TAPE11	-	Intermediate file used in merging layers; label alternates between LFILE and MFILE file structure - buffered binary format.
TAPE12	-	Final output file but used in intermediate merging of layers; indexing set to assure final result is on this unit; label may alternate between LFILE and MFILE but always MFILE=12 for final layer buffered binary format.

<u>UNIT LABELS:</u>

KFILE	-	Contains HIRACV output for current layer - buffered binary
LFILE	-	Contains results of previous layer - buffered binary
MFILE	-	Contains merged results from KFILE and MFILE - buffered binary

capability on the machine being used, allowance will have to be made for a larger amount of central memory. In this case, storage allocation should be redistributed to minimize total storage requirement, estimated at 77 K_8. That is, long extension arrays in EMUP, EMDWN, and ABS should be changed to unlabelled common.

A simple overall structure of the code was obtained. This is shown schematically in Figure A.1. HIRACV and its associated subroutines perform the spectrum synthesis of the absorptance. The merging subroutine is called ABSMRG for the absorption coefficient case. The LTE radiance computations are performed by subroutines EMINIT, EMUP, and EMDOWN. Subroutine TPLOT is an expanded version of the plotting program described in Reference [3], and it has been changed into subroutine form. Note that FASCODE can be used to prepare plots directly. Some possible future extensions of the program have been sketched as dotted lines in Figure A.1. The reader should note that the modular construction of the code allows straightforward extension and revisions.

The program consists of the main Program FASCODE: Subroutines ABSMRG, TPLOT, EMINIT, EMUP, EMDOWN, HIRACV, SHAPED, SHAPEL, MOLEC, RDFILE, CONVFNV, PANEL, HIRACL, CONVFNL, HIRACD, CONVFNC, and PANELD; and the Function QVRFAC. The overall strategy of the HIRACV Subroutine is indicated in Figure A.2. All the subroutines are called from the main program or from their subdriver (See Figure A.1) and the flow of the program is easily traced. Subroutines HIRACL and HIRACD perform the same functions for the Lorentz and Doppler line shape profiles respectively.

The main Program FASCODE reads the input data and calls each of the subdrivers shown in Figure A.1, according to the particular run desired as determined from the input. After initializing constants and reading the basic parameters for the run, the program enters a loop for the calculation of the requested results for each layer. Inside the loop the atmospheric properties, average temperature, average pressure and absorber molecule column densities are read from the input file for the given layer. The proper sampling interval is then computed and an identification header for the layer is buffered out to KFILE. Note that extensive use of BUFFER IN and BUFFER OUT is made to increase efficiency. If a user does not have this

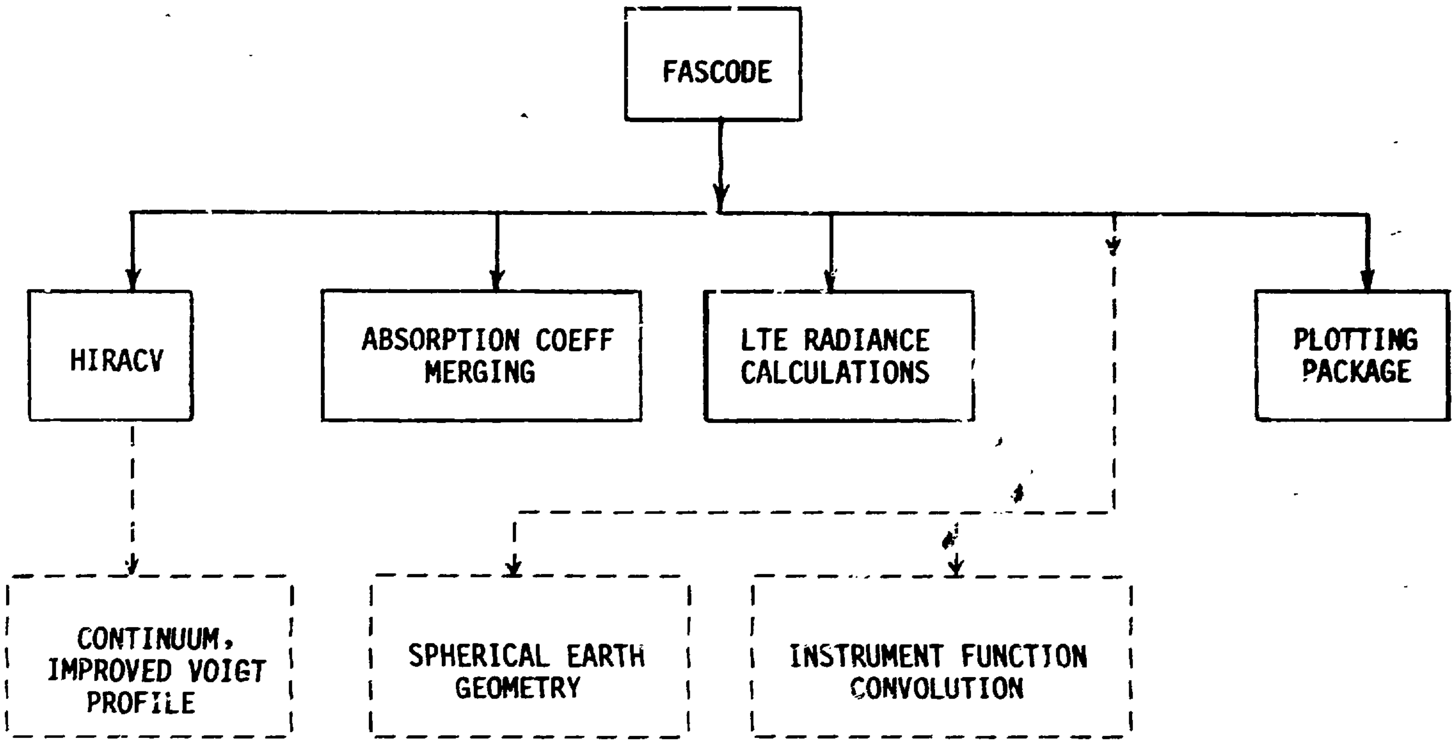

Figure A.1: FASCODE Overall Structure.

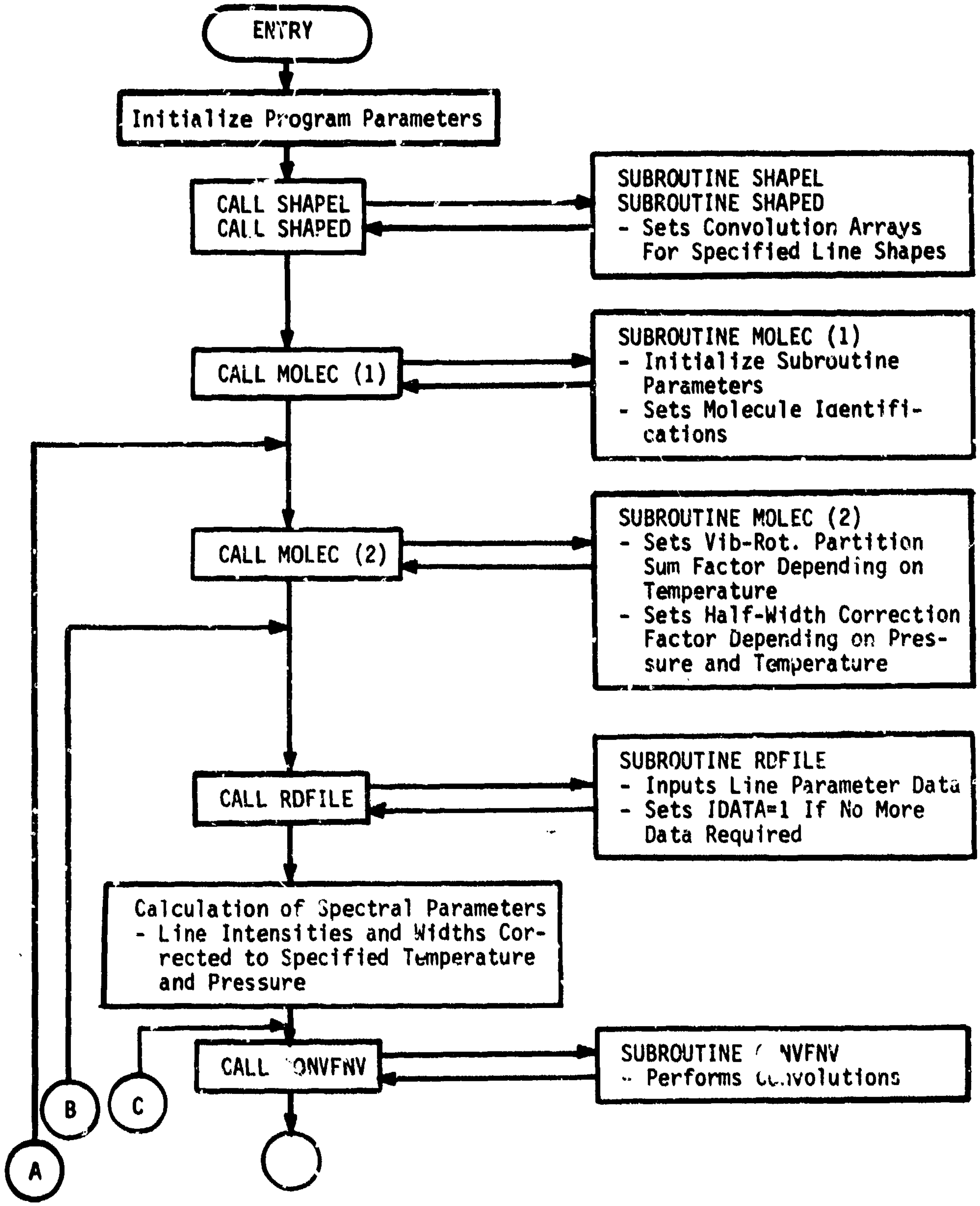

Figure A.2: Flow Diagram for HIRACV Subroutine

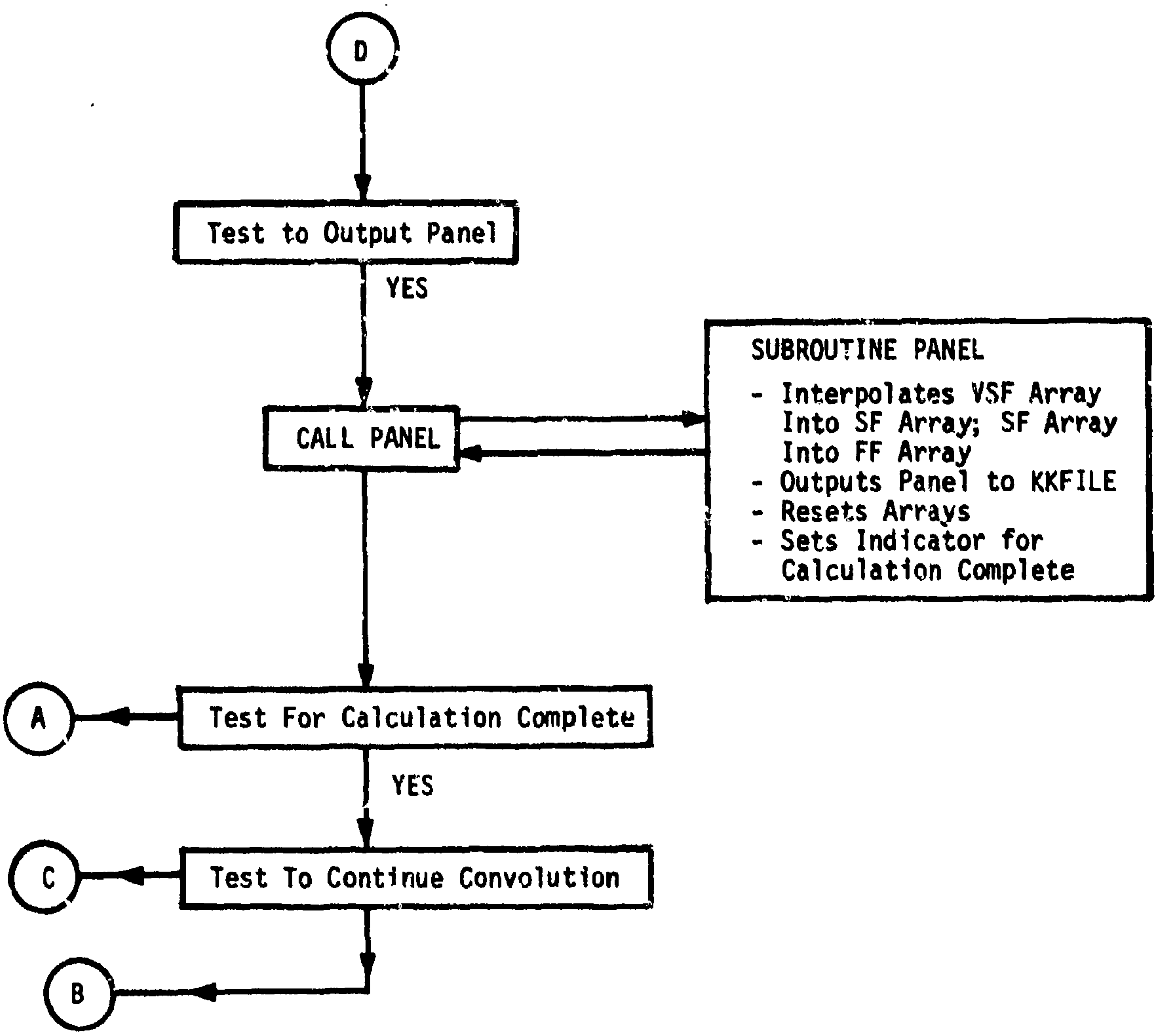

Figure A.2: Flow Diagram for HIRACV Subroutine (Continued) capability, these statements can be replaced by binary read and write statements. Subroutine HIRACV is then accessed to compute the spectral absorptance. Once this has been completed, the appropriate merging routine is accessed, unless it is the first layer for which no merger is necessary. For the radiance calculation, Subroutine EMINIT is called to compute the radiance due to the first layer. For all other layers, the radiance is obtained by calls to EMDOWN or EMUP for the cases of looking down and looking up respectively. When all layers have been processed, the plotting Subroutine TPLOT is called if the plot flag has been set by the user. Subroutines SHAPEL and SHAPED set up the convolution functions used to define the Voigt function from 0 to 64 half-widths.

Subroutine MOLEC in conjunction with Function QVRFAC, makes the molecular identifications associated with the line parameter file, and determines the correction factors for the line intensities (SCOR) and the half-widths (ALFCOR). The quantity, SCOR, is the correction factor due to the temperature dependence of the vibrational and rotational partition sums. The vibrational partition sum is calculation for a given molecular type as

$$Q_v(T) = \prod_{i=1}^{N} \frac{1}{\left(1 - e^{-h\nu_i/kT}\right)^{d_i}} \tag{A-1}$$

where ν_i is a fundamental frequency and d_i is the degeneracy of the vibration. The temperature dependence of the rotational partition sum is given by

$$\frac{Q_R(T_o)}{Q_R(T)} \simeq \left(\frac{T_o}{T}\right)^F \tag{A-2}$$

where F = 1 for linear molecules and 1.5 for nonlinear molecules. The reference temperature, T_o, is taken as 296 K, consistent with the AFGL Line Listing. For further discussion of these topics, see Herzberg[21] pp. 503 ff. The partition sum calculations are performed by QVRFAC and the necessary molecular parameters are contained in data statements in Subroutine MOLEC. The quantity, ALFCOR, is the correction factor due to the pressure and temperature dependence of the collision broadened half-width. The temperature dependence of the half-width has been taken as $(T_0/T)^{0.5}$ although calculations based on the Anderson-Tsao-Curnutte theory are reported to give somewhat different temperature dependencies (Varanasi)[22].

Subroutine RDFILE reads the blocked binary line parameters over the wave number range for which line data is required. The line parameters include the wave number value of the transition (GNU, cm^{-1}), the intensity of the transition at 296°K (S, $cm^{-1}/(mol/cm^2)$), the collision broadened half-width at half maximum for 296°K and 1 atm pressure (ALFAO, cm^{-1}), the lower state energy of the transition (EPP, cm^{-1}), and the molecule identification number (MOL). If the line parameter data is insufficient to complete the specified calculation, the message "end of file on disk" is printed on the output file. If no further line data is required, IDATA is set to 1, and control is returned to the main program.

At this stage of the main program, an effective optical depth is calculated for each line which is dependent on the column density of the layer, the secant of the angle through the layer, the temperature of the layer, and the half-width of the line α(ALFI).

The effective depth, μ',

$$\mu' = \left(\frac{S(T)}{\alpha_V(T)}\right) \cdot w \cdot \sec \cdot \left(\frac{Q_{VR}(T_o)}{Q_{VR}(T)}\right) \cdot \left[\exp\left(\frac{E''}{kT_o} - \frac{E''}{kT}\right)\right] \cdot \left[\frac{1 - \exp\left(-\frac{h\nu}{kT}\right)}{1 - \exp\left(-\frac{h\nu}{kT_o}\right)}\right] \qquad \text{(A-3)}$$

where w is the absorber column density, E" is the lower state energy, $Q_{VR} \equiv Q_V Q_R$, and the other quantities have been previously defined. In terms of the program coding the effective depth appears as:

```
EFDPTH = SEC*W*S*SCOR*RECALF*
                EXP(EPP/XKTFAC) + (1.-EXP(-GNU/XKT))/(1.-EXP(-GNU/XKTO))
```

where XKT and XKTO are the wave number values of T and T_0 in cm^{-1}. As previously discussed, the proper sampling interval, DV, should be 0.25 times the average line half-width. If the half-width, ALFI, is less than the sampling interval, the half-width is set to the sampling interval and a series of minus signs is written to the output file. If the half-width exceeds a maximum value (ALFMAX) where ALFMAX=BOUND/64 and BOUND is the maximum value in wave numbers over which a line can be calculated, the half-width is reset to ALFMAX, and a series of + signs is written to the output file. Included in the records indicating the resetting of the half-width is the wave number value of the transition (GNU), the intensity (S) and the half-width (ALFAO) values of the transition from the line parameter tape, the calculated value of the half-width (ALFI), the value to which the half-width has been reset (DV or ALFMAX), and molecular identification number (M). If the number of half-width changes (NCHNG) exceeds 100, the computation is terminated.

Subroutine CONVFNV is a tightly written subprogram in which considerable effort has been taken to minimize operations in the DO 30 loop. This subroutine performs the triple convolution of XF, XS, and XVS with a line datum putting the results in the proper elements of FF, SF, and VSF respectively. A simplified flow diagram appears in Figure A.3. Control indicator IPANEL is set to IDATA if the DO loop over the lines (40) is satisfied indicating whether a panel is complete or more lines are required. If the line DO loop (40) is not completed, IPANEL is set to 1 indicating that a complete panel has been calculated. Control is returned to the main program.

If IPANEL has been set to 1, Subroutine PANEL is called. Subroutine PANEL performs a four-point Lagrange interpolation of the VSF array into the SF array and the SF array into the FF array, thus combining the results of three independent convolutions into a final result. A general flow diagram of PANEL is given in Figure A.4. Care is taken to store array values required for the interpolation of subsequent panels. VFT is the wave number value of the first element of the FF array, which is common to the first element of the SF and VSF arrays. A binary header record is written to the binary file (KFILE) for each panel which includes the wave number value (V1P) of the first element of the panel (FF(NLO)), the wave number value (V2P) of the last element of the panel (FF(NHI)), the wave number increment (DV), and the number

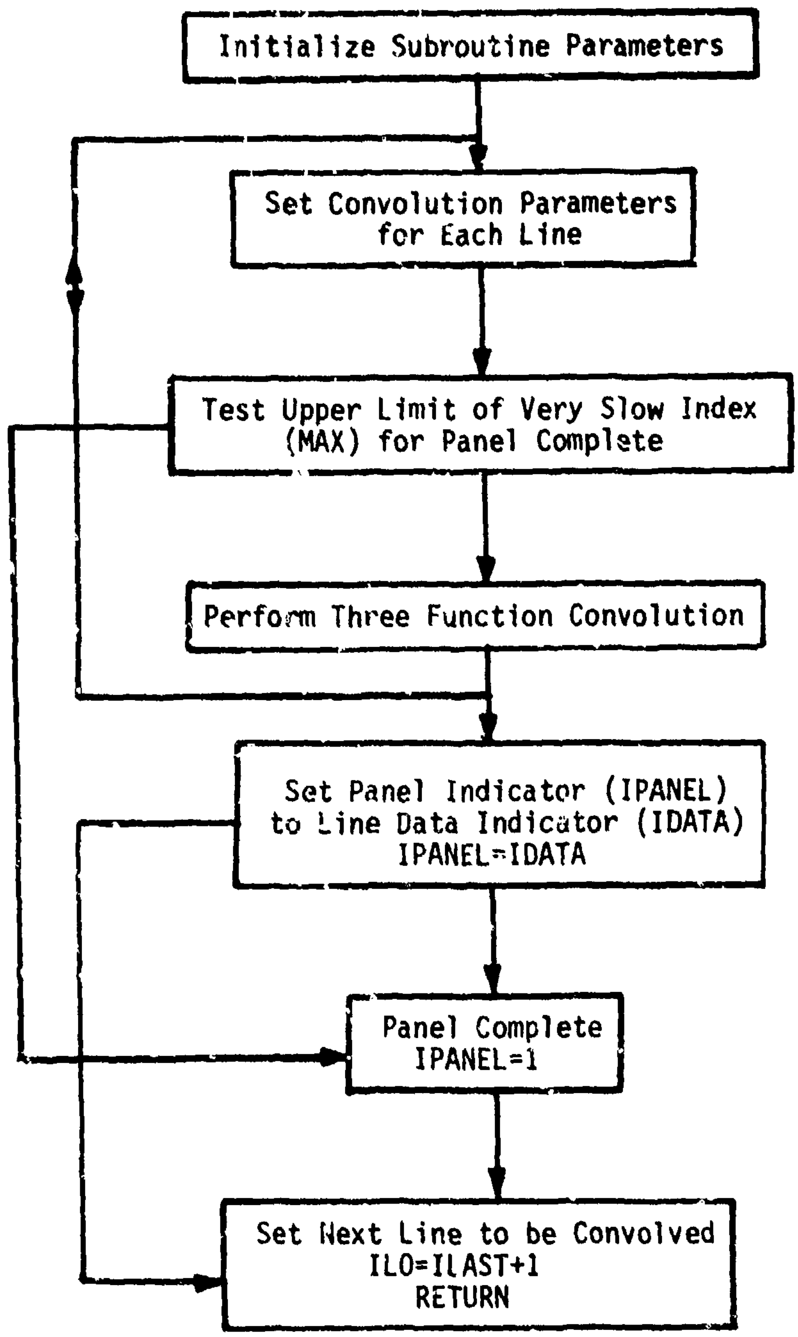

Figure A.3: Flow Diagram for SUBROUTINE CONVFNV.

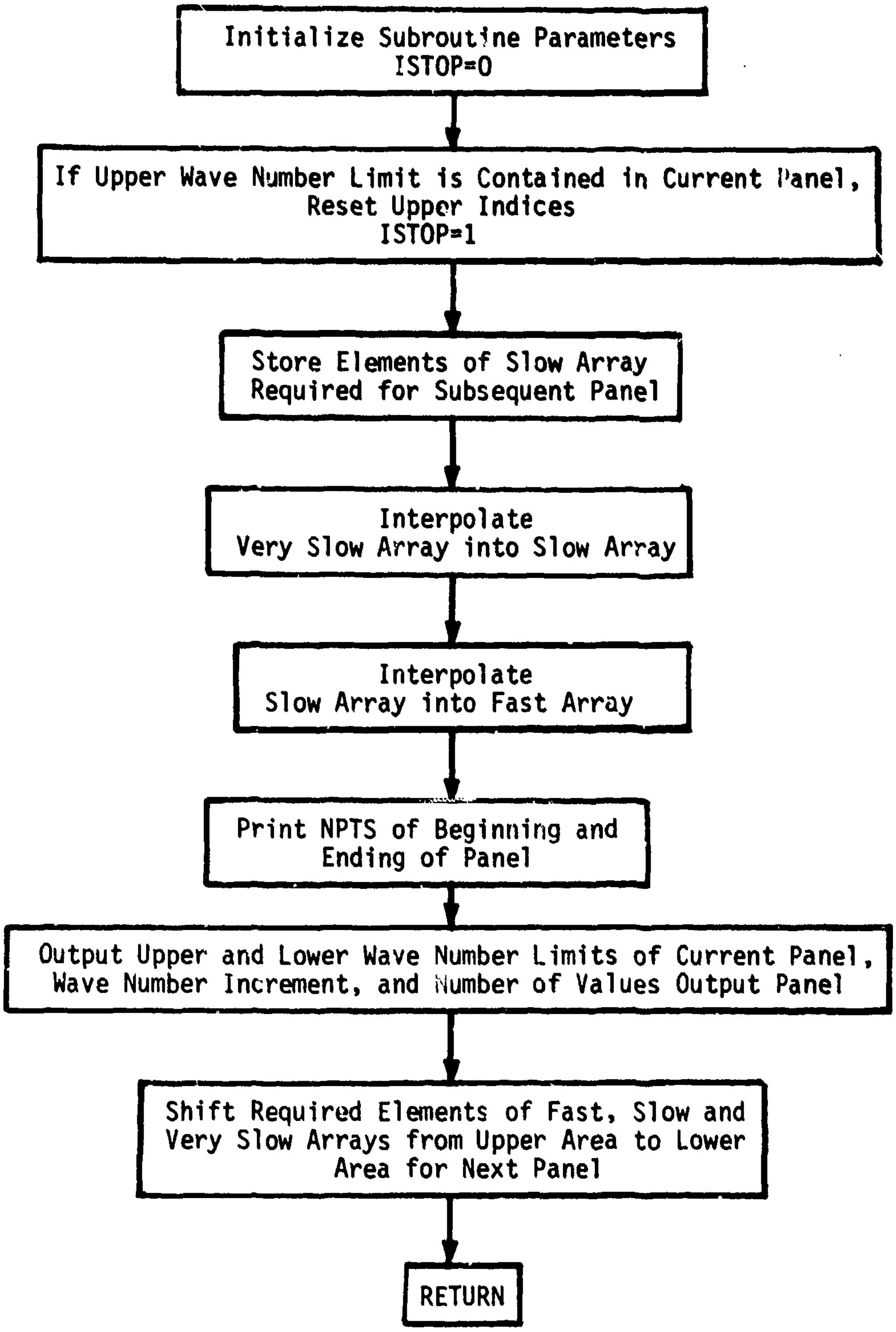

Figure A.4: Flow Diagram for SUBROUTINE PANEL.

of absorption coefficient values outputted (NLIM). The second binary record contains the NLIM values of the absorption coefficient from the FF array. The arrays are appropriately shifted and reset in preparation of the computation of subsequent panels. Control is again returned to the main Subroutine HIRACV.

At the conclusion of the outputting of the last panel for each layer, a record is written to the output file indicating the current value of the time, the time spent in RDFILE, in CONVFN, and in PANEL (the units are seconds). Also included in this record are the first and last wave number values of the panel. A second record is written to the output file indicating the average value of the half-width, the number of lines read since the last panel was completed, and the total number of lines read since the initiation of the convolution calculation. Control is returned to Statement 10 if the calculation is complete, or to Statement 40 to continue the convolution in process.

ABSMRG -- Absorption Merging

Subroutine ABSMRG performs the merging of the absorption layer-by-layer along a path. The absorption at the present layer and the accumulated absorption from previous layers are read from disk files. The constant ISMALL, computed in FASCODE, is used to identify whether the present or the accumulated layer has the smaller wave number increment. BUFFER IN and BUFFER OUT are used rather than READ and WRITE for speed of running. The array FF contains the absorption from the smaller increment and DUMF, the absorption from the larger. The index ITYPE, computed in FASCODE, gives the value of the numerator of the ratio of the two increments. The coefficients for the Lagrange four-point interpolation are computed and an interpolation is made for the absorption in the layer with the larger increment. The merging of the two layers is then carried out such that the merged absorptance has an increment equal to the smaller sampling interval. The output file contains a header record containing identification information, the values of the secant, pressure and temperature concentrations of the absorbing molecules, wave number increment, the value of the first, and last wave number in the panel and a layer count. Each panel's output consists of two records. The first record contains the first and last wave number in the panel, the wave number increment and the number of points in the panel. The second record contains the accumulated absorption. For speed of computation, if the wave number increment is the same for both layers, the merging is done directly by merely adding the corresponding values. A simplified flow diagram is shown in Figure A.5.

Subroutine EMINIT computes the LTE radiance for the first altitude layer only. The Planck function is computed as described in the text. There is also the option of the radiance from a boundary, for the case of looking down. A simplified flow chart is given in Figure A.6. The radiance for the case looking from the space to ground is computed by the formula

NEWEM = (1 - TR)*BB + TR*OLDEM

The radiance from ground to space uses the formula

NEWEM = (1 - TR)*BB

where TR is the incremental transmission of the layer, BB the Planck function and OLDEM the radiance from the boundary when it is requested. Otherwise, it is zero. The results of the calculation are written on a file consisting of four records:

1. A header record containing alpha-numeric information supplied from FASCODE in the first seven words, the values of the secant, average pressure, average temperature, concentrations of the absorbing molecules, DV, first and last wave numbers and a layer count.
2. V1, V2, DV and the number of points in the panel.
3. The radiance.
4. The transmission

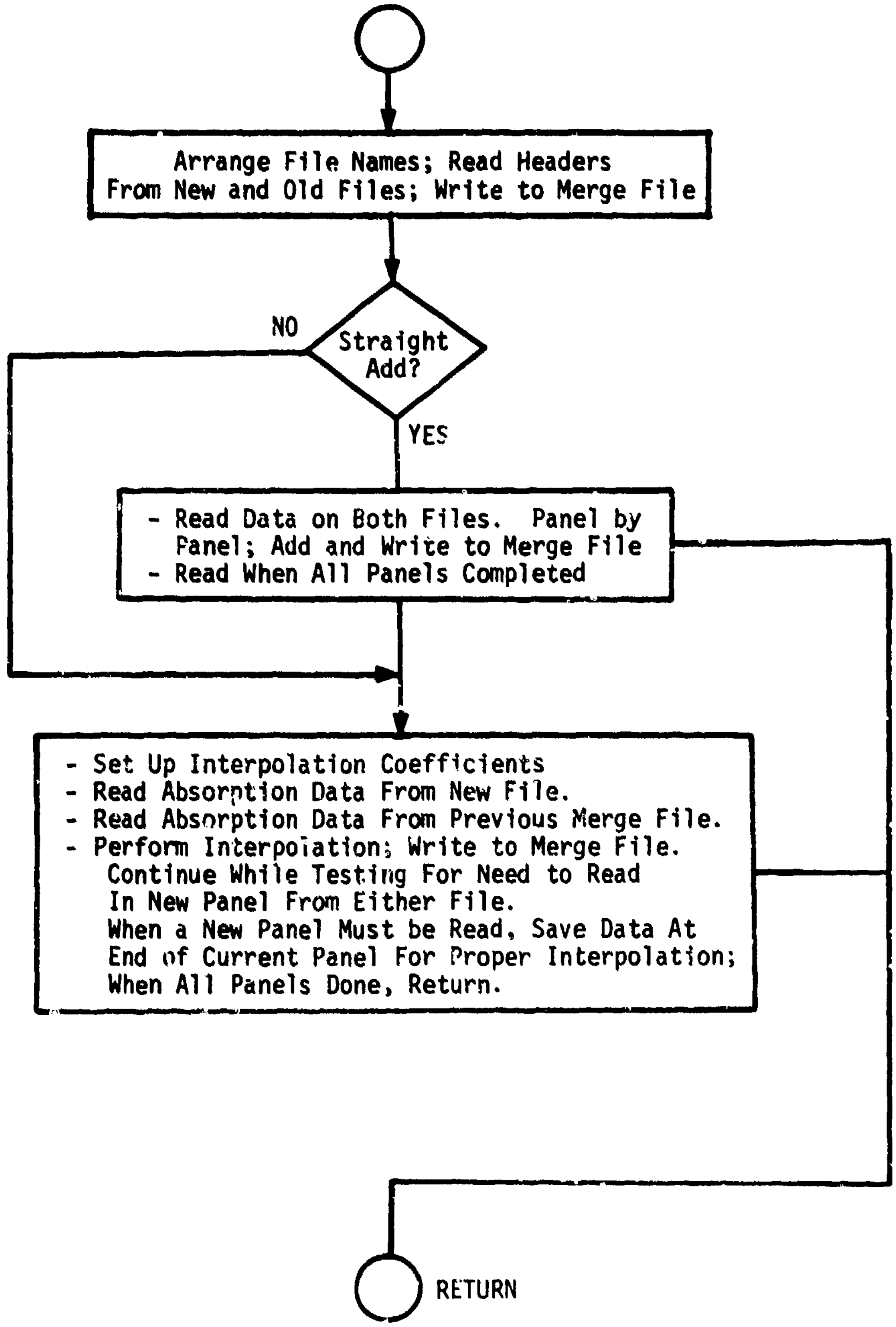

Figure A.5: Flow Diagram for SUBROUTINE ABSMRG.

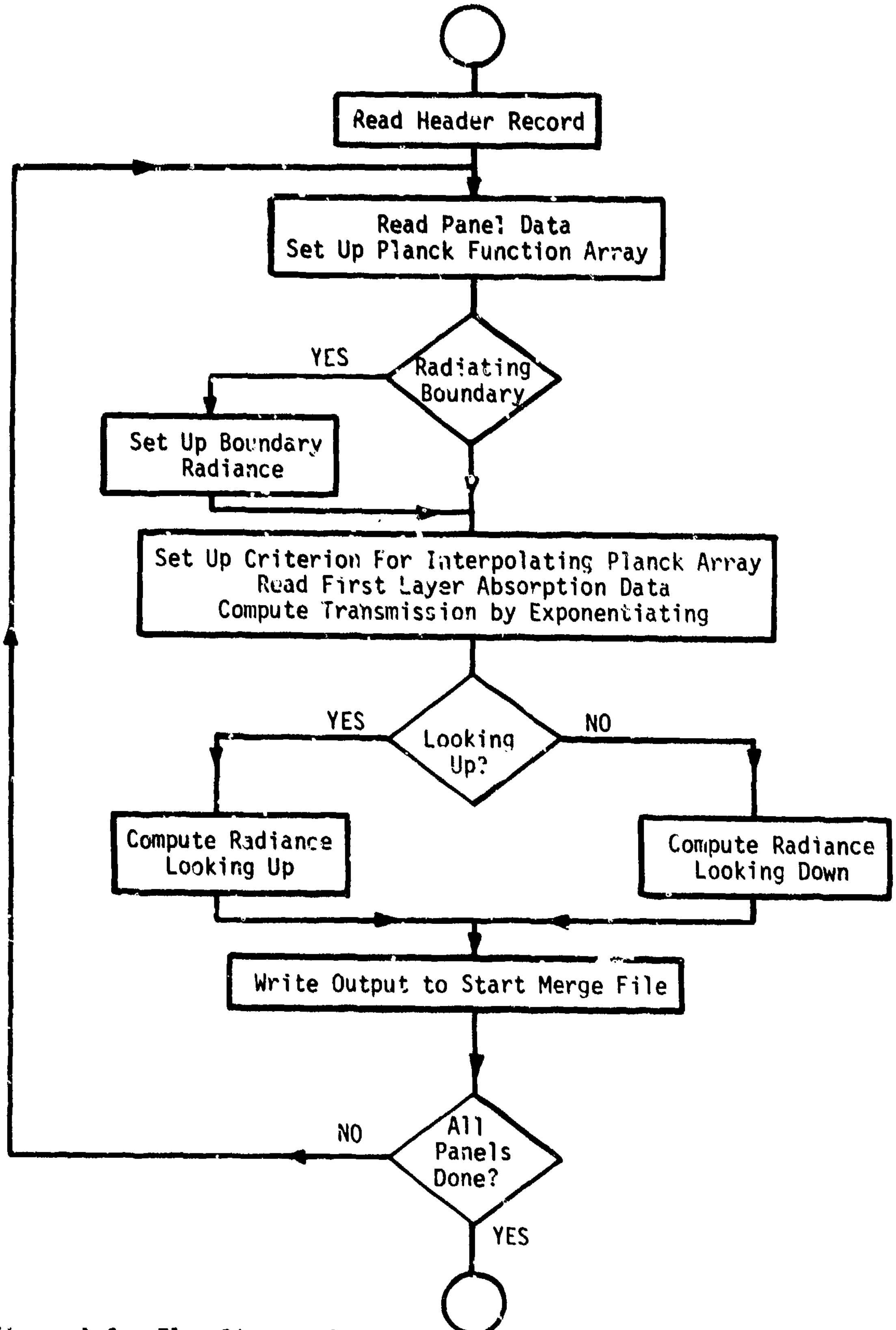

Figure A.6: Flow Diagram for EMINIT

EMUP -- Radiance and Transmittance Looking Up

Subroutine EMUP performs the merging of the radiance and transmission layer-by-layer along a path looking up from ground to space. As in Subroutine ABSMRG, the 1/1 ratio of the DV's (wave number increment) is done separately for speed of computation. For the remaining ratios, ITYPE is computed in FASCODE and equals the value of the numerator of the ratio of the two DV's. The coefficients for the Lagrange four-point interpolation are computed and a file is read which contains the information for the accumulated quantities. This file consists of a header record which includes identification information, the value of the secant, pressure and temperature, concentrations of the absorbing molecules, wave number increment, the value of the first and last wave number in the layer, and a layer count. This is followed by a series of three records per panel. The first record contains the first and last wave number in the panel, the wave number increment and the number of points in the panel. The second record contains the accumulated radiance which is stored in array OLDEM. The third, the accumulated transmission stored in array OLDTR.

Similarly, the file for the new layer contains the header record and first and last wave number record. The next record contains the spectral absorptance from which the transmission is computed and is stored in array TR.

The black body function is computed as described in the text and an interpolation is made for the radiance and transmission with the larger wave number increment. The total radiance along the path length is computed using the formula

$$\mathrm{NEWEM=OLDEM} + (1.0 - \mathrm{TR}_i)*\mathrm{BB}_i*\mathrm{OLDTR}$$

where OLDEM is the interpolated old radiance, TR_i the new transmission increment, BB_i, the black body and OLDTR the interpolated transmission. The transmission is computed as

$$\mathrm{NEWTR=OLDTR}*\mathrm{TR}_i$$

The output file is written such that it can be used as the input when treating the next layer. The flow chart for Subroutine EMUP is in Figure A.7.

EMDOWN -- Radiance and Transmittance Looking Down

Subroutine EMDOWN performs the merging of the radiance and transmission layer-by-layer along a path looking down from space toward ground. However, to achieve efficiency it is arranged such that the calculation always starts at the lowest altitude and continues upward. The logic for EMDOWN is exactly the same as Subroutine EMUP and the formula for computing transmission is the same. The LTE radiance, however, is computed as follows:

$$NEWEM=(1 - TR_i)*BB_i + TR_i*OLDEM$$

where OLDEM is the interpolated old radiance, TR_i the transmission increment of the i'th layer and BB_i the black body function. In looking from space to ground it is not strictly necessary to compute the total transmission in order to compute the radiance but this feature has been included in this subroutine to make it compatible with Subroutine EMUP. Figure A.7 also gives the tiow diagram for this routine.

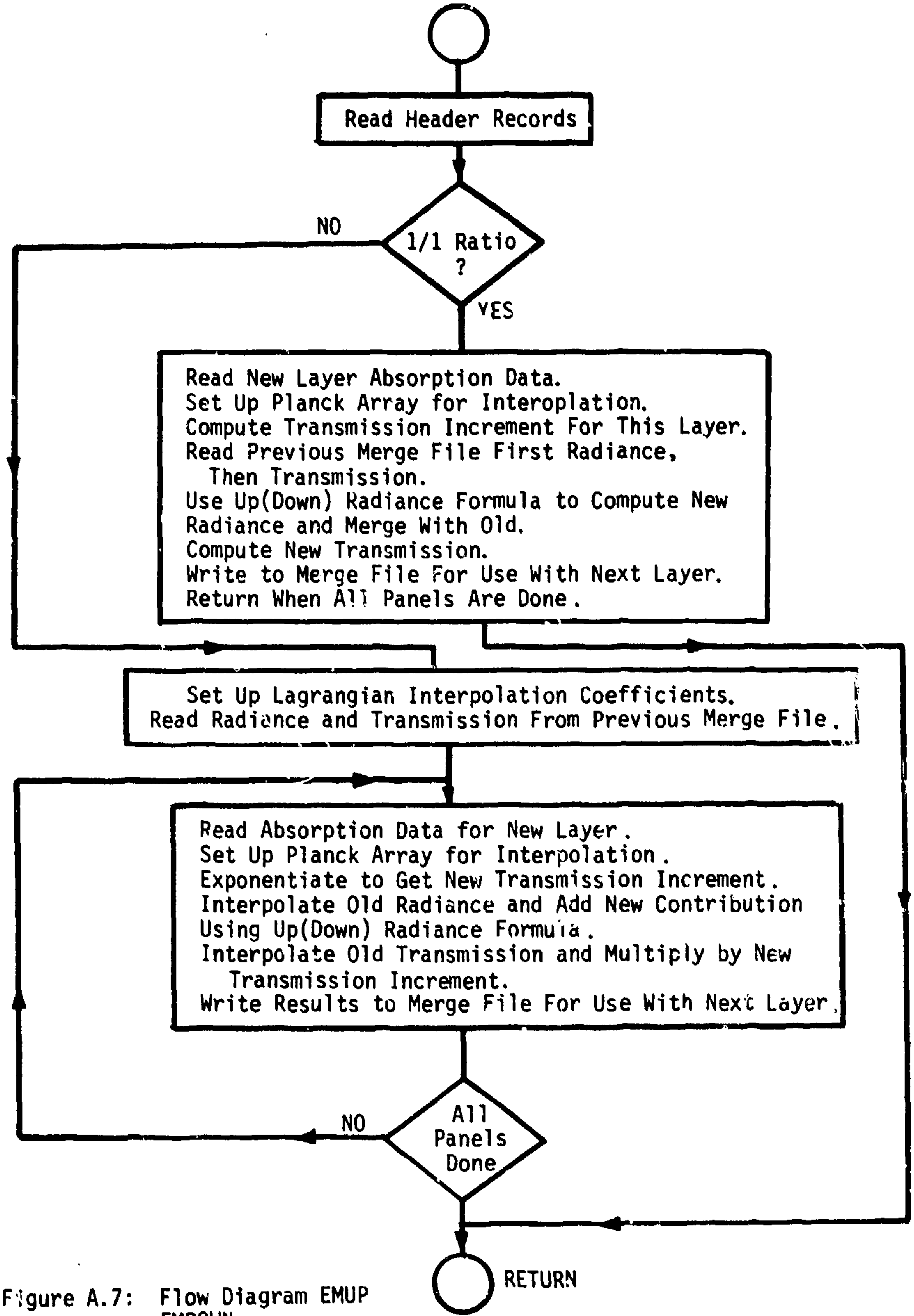

Figure A.7: Flow Diagram EMUP EMDOWN

APPENDIX B

USER'S GUIDE

Appendix B presents samples of input and output from a run of FASCODE on the AFGL CDC 6600 computer. A user may find these pages helpful in assessing whether the program has been adapted properly to a new machine environment.

As pointed out already, extensive use has been made of the BUFFER IN and BUFFER OUT capability available in CDC FORTRAN Extended. Since other computer systems (such as IBM 360/370) may not have this feature available, we have collected a list of all the BUFFER statements together with their identification numbers. This is given in Table B.1 and should be of assistance to a user whose machine requires the changing of the BUFFER statements to binary READ and WRITE statements.

Table B.2 gives a sample job setup for a run used in developing Test Case Output. The first record contains CDC job control cards (SCOPE). This is followed by a series of Segmentation Loader input cards which determine the way in which FASCODE is to be loaded into the system. The third and final record contains the input data for the execution of the Test Case Problem by FASCODE. Table B.3 contains a list and description of the FASCODE input data. A load map for this run is found in Table B.4. Finally, the output printed by FASCODE in executing this job is presented in Table B.5. To ascertain that a faithful version is in hand, it is recommended that a user execute this test case and compare the output with Table B.5.

With regard to the preparation of model atmospheres for input to FASCODE, the principal requirement is to assure that ratios of the sampling intervals between a layer and the one just above it is less than or equal to 2/1. It is suggested that 3/2 would be a good target ratio. This can be checked by computing the average half-width of the layers using Equations (2.2) or (2.4). If a user would feel more comfortable in his mind that a good model has been derived, one or two extra layers could be added at relatively low cost. However, it should be clear that each new layer increases the running time correspondingly.

TABLE B.1. LOCATION OF BUFFER STATEMENTS

	BUFFER IN(LINFIL,1) (FILHD(1),VLINHI)	000630
	BUFFER IN (LINFIL,1) (FILHD(1),VLINHI)	000790
	BUFFER OUT(KFILE,1) (XID(1),NLAYER)	002540
10	BUFFER IN (LINFIL,1) (VMIN,NREC)	003660
	BUFFER IN (LINFIL,1) (GNU(I1),GNU(I2))	003710
30	BUFFER IN (LINFIL,1) (GNU(I1),MOL(LIMIN))	003750
	BUFFER OUT (KFILE,1) (V1P,NLIM)	005250
	BUFFER OUT (KFILE,1) (FF(NLO),FF(NHI))	005270
	BUFFER IN(KFILE,1)(XID(1),XID(2))	009040
260	BUFFER OUT(MFILE,1)(XID(1),NLAYER)	009060
420	BUFFER IN(LFILE,1)(XID(1),NLAYER)	009080
	BUFFER IN(IOLD,1)(OV1P,NLIMO)	009190
	BUFFER IN(IOLD,1)(DUMF(1),DUMF(NLIMO))	009210
	BUFFER IN(KNEW,1)(V1P,NLIM)	009230
	BUFFER IN(KNEW,1)(FF(1),FF(NLIM))	009250
	BUFFER OUT(MFILE,1)(V1P,NLIM)	009290
	BUFFER OUT(MFILE,1)(FF(1),FF(NLIM))	009310
	BUFFER IN(IOLD,1)(OV1P,NLIMO)	009550
310	BUFFER IN(IOLD,1)(DUMF(1),DUMF(NLIMO))	009580
	BUFFER IN(KNEW,1)(V1P,NLIM)	009620
330	BUFFER IN(KNEW,1)(FF(1),FF(NLIM))	009650
	BUFFER IN(IOLD,1)(OV1P,NLIMO)	009920
	BUFFER IN(IOLD,1)(DUMF(4),DUMF(NLIMO))	009970
150	BUFFER OUT(MFILE,1)(V1P,NLIM)	010170
370	BUFFER OUT(MFILE,1)(FF(1),FF(NLIM))	010190
160	BUFFER IN(KNEW,1)(V1P,NLIM)	010250
350	BUFFER IN(KNEW,1)(FF(1),FF(NLIM))	010280
	BUFFER IN(KFILE,1)(XID(1),NLAYER)	010720
	BUFFER OUT(MFILE,1)(XID(1),NLAYER)	010740
	BUFFER IN(KFILE,1)(OV1P,NLIMO)	010780
	BUFFER IN(KFILE,1)(FF(1),FF(NLIMO))	011[illegible]50
	BUFFER OUT(MFILE,1)(OV1P,NLIMO)	011360
	BUFFER OUT(MFILE,1)(EMISS(1),EMISS(NLIMO))	011330
	BUFFER OUT(MFILE,1)(FF(1),FF(NLIMO))	011400
	BUFFER IN (LFILE,1) (XID(1),XID(2))	011780
	BUFFER IN(KFILE,1)(XID(1),NLAYER)	011800
	BUFFER OUT(MFILE,1)(XID(1),NLAYER)	011820
535	BUFFER IN(LFILE,1)(OV1P,NLIMO)	011950
	BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))	011970
	BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))	011990
	BUFFER IN(LFILE,1)(OV1P,NLIMO)	012420
	BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))	012480
	BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))	012530
	BUFFER IN(LFILE,1)(OV1P,NLIMO)	013060
	BUFFER IN(LFILE,1)(OLDEM(4),OLDEM(NLIMO))	013100
130	BUFFER IN(LFILE,1)(OLDTR(4),OLDTR(NLIMO))	013130
150	BUFFER OUT(MFILE,1)(V1P,NLIM)	013420
	BUFFER OUT(MFILE,1)(NEWEM(1),NEWEM(NLIM))	013440
	BUFFER OUT(MFILE,1)(NEWTR(1),NEWTR(NLIM))	013460
360	BUFFER IN(KFILE,1)(V1P,NLIM)	013630
	BUFFER IN(KFILE,1)(TR(1),TR(NLIM))	013650
	BUFFER IN(LFILE,1) (XID(1),XID(2))	014570
	BUFFER IN(KFILE,1)(XID(1),NLAYER)	014590
	BUFFER OUT(MFILE,1)(XID(1),NLAYER)	014610
535	BUFFER IN(LFILE,1)(OV1P,NLIMO)	014740
	BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))	014760
	BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))	014780
	BUFFER IN(LFILE,1)(OV1P,NLIMO)	015220
480	BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))	015280
	BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))	015300
	BUFFER IN(LFILE,1)(OV1P,NLIMO)	015800
	BUFFER IN(LFILE,1)(OLDEM(4),OLDEM(NLIMO))	015840
42	BUFFER IN(LFILE,1)(OLDTR(4),OLDTR(NLIMO))	015860
150	BUFFER OUT(MFILE,1)(V1P,NLIM)	016150
	BUFFER OUT(MFILE,1)(NEWEM(1),NEWEM(NLIM))	016170
	BUFFER OUT(MFILE,1)(NEWTR(1),NEWTR(NLIM))	016190
360	BUFFER IN(KFILE,1)(V1P,NLIM)	016390
	BUFFER IN(KFILE,1)(TR(1),TR(NLIM))	016410

TABLE B.2. SAMPLE RUN DECK FOR TEST PROBLEM

```
 74/74    OPT=1                              FTN 4.6+425        03/14/78  16.43.20

CLOUGH,CM57700,T48.            FASCODEL             2  4     CLOUGH
ATTACH,TAPE3,CLOU1990TO2210,ID=CLOUGH,MR=1.
REQUEST,TAPE12,*PF.
ATTACH,HIRACL,ID=CLOUGH,MR=1.
FTN,I=HIRACL,SL,B=LGO.
ATTACH,HIRACD,ID=CLOUGH,MR=1.
FTN,I=HIRACD,SL,B=DGO.
ATTACH,VGO,FASCODE,ID=CLOUGH,MR=1.
REWIND,OUTPUT.
MAP,PART.
LDSET,PRESET=INDEF.
LOAD,VGO.
LOAD,LGO.
LOAD,DGO.
SEGLOAD.
EXECUTE,FASCODE.
EOR
FAST TREE FASCODE-(HIRACV,HIRACL,HIRACD,ABSMRG,EMINIT,EMUF,EMOONN)
FASCODE GLOBAL MAIN,NEW
HIRACV INCLUDE SHAPEL,SHAPED,VOICON,RDFILE,CONVFNV,PANEL,MOLEC,OVRFAC
HIRACL INCLUDE SHAPEL,RDFILE,CONVFNL,PANEL,MOLEC,OVRFAC
HIRACD INCLUDE SHAPED,RDFILE,CONVFNC,PANELD,MOLEC,OVRFAC
       END FASCODE
EOR
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER
   1    1    0    0    0
 1.
 2095.    2105.
 273.
  10
 525.8566  264.3769                        4.00 KM TO    7.00 KM
9.380E+21 1.409E+21 2.611E+17 1.195E+18 3.201E+17 5.831E+18 8.942E+23 3.334E+24
 332.8741  242.7476                        7.00 KM TO   11.00 KM
1.649E+21 1.284E+21 4.587E+17 1.089E+18 2.917E+17 5.724E+18 8.149E+23 3.037E+24
 197.5644  222.8735                       11.00 KM TO   14.00 KM
6.815E+19 6.262E+20 5.351E+17 5.313E+17 1.423E+17 3.035E+18 3.975E+23 1.482E+24
 124.0100  216.0000                       14.00 KM TO   17.00 KM
7.612E+18 4.043E+20 7.852E+17 3.429E+17 9.185E+16 1.960E+18 2.566E+23 9.543E+23
  77.2771  216.9193                       17.00 KM TO   20.00 KM
5.045E+18 2.509E+20 1.081E+18 2.129E+17 5.702E+16 1.216E+18 1.593E+23 5.936E+23
  36.5033  223.8896                       20.00 KM TO   30.00 KM
1.349E+19 3.284E+20 3.311E+18 2.786E+17 7.462E+16 1.592E+18 2.085E+23 7.772E+23
   8.3402  242.9639                       30.00 KM TO   40.00 KM
4.989E+18 7.053E+19 1.259E+18 5.982E+16 1.603E+16 3.419E+17 4.477E+22 1.663E+23
   2.5509  263.2770                       40.00 KM TO   45.00 KM
4.913E+17 1.118E+19 1.529E+17 9.484E+15 2.541E+15 5.425E+16 7.097E+21 2.645E+22
   1.1559  267.2521                       45.00 KM TO   54.00 KM
2.590E+17 8.437E+18 6.597E+16 7.156E+15 1.917E+15 4.090E+16 5.355E+21 1.996E+22
    .2742  255.4215                       54.00 KM TO  100.00 KM
4.745E+16 4.080E 18 1.176E 16 3.464E 15 9.276E 14 1.979E 16 2.591E 21 9.649E 21
```

TABLE B.3. INPUT DATA DESCRIPTION

XID	7 element array containint text. The text is printed with FASCODE output and is used as a header for execution time plots.
IHIRAC	1, Voigt approximation to line shape 2, Lorentz approximation to line shape 3, Doppler approximation to line shape
IEMIT	0, no emission calculation 1, emission calculation performed
ILOOK	0, emission calculation from ground to space 1, emission calculation from space to ground
IPLOT	0, no plot 1, on-line plot 2, off-line plot 3, CRT plot 4, graphics plot
ISCAN	indicates parameter to be plotted (use if IPLOT≠0) 1, absorption coefficient YY=YY 2, log of absorption coefficient YY=ALOG10(YY) 3, transmission YY=exp(-YY)
SECANT	secant of the angle between the line-of-sight and the zenith
V1 (lower limit) V2 (upper limit)	specify the wave number range for the output data (cm^{-1})

TABLE B.3. INPUT DATA DESCRIPTION (continued)

TBOUND	temperature (°K) used to compute a boundary. If TBOUND=0, no boundary is computed (currently used for looking down)
NNLAYR	number layers + 1
PAVE	average pressure (mb) for the layer
TAVE	average temperature (°K) for the layer
WK	array containing the molecular column densities for the layer (molecules/cm^2) H_2O, CO_2, O_3, N_2O, CO, CH_4, O_2

****

IF IPLOT=1

PLOTID	3 element array containing 30 characters of text used as a beginning and ending banner on pen plots
V1 V2	the range of wave numbers over which plot will made (cm^{-1})
XSIZE	size of X axis (in)
DX	number of units of X per inch
YMIN YMAX	range of Y-axis
WWSC	the constant k in the expression $y=e^{-ky}$

TABLE B.4. LOAD MAP

```
*FASCODE
 ?
 ?_HIRACV
 ?
 ?_HIRACL
 ?
 ?_HIRACD
 ?
 ?_ABSMRG
 ?
 ?_EMINIT
 ?
 ?_EMUP
 ?
 ?_EMDOWN
READING FILE        VGO
          PROGRAM   FASCODE L=012305  BLOCKS            CM015477  MAIN    CM000004  NEW     CM000035  BUFHD   CM000040
          PROGRAM   HIRACV  L=002735  BLOCKS            CM015477  MAIN    CM000004  NEW     CM000035  XSUB    CM000012
                                      BLOCKS  WXX       CM000014  SUB1    CM000012  XPANEL  CM000006  XTIME   CM000004
                                      BLOCKS  VOICOM    CM000622
          PROGRAM   RDFILE  L=000163  BLOCKS  XSUB      CM000012  MIF2    CM000003
          PROGRAM   CONVFNV L=000247  BLOCKS  NEW       CM000035  XSUB    CM000012  WXX     CM000014  SUB1    CM000012
                                      BLOCKS  XTIME     CM000004  VOICOM  CM000622
          PROGRAM   PANEL   L=006503  BLOCKS  NEW       CM000035  XSUB    CM000012  SUB1    CM000012  XPANEL  CM000006
                                      BLOCKS  XTIME     CM000004
          PROGRAM   MOLEC   L=000252  BLOCKS  XMOLEC    CM000124
          PROGRAM   OVRFAC  L=000111
          PROGRAM   SHAPEL  L=000266  BLOCKS  WXX       CM000014
          PROGRAM   SHAPED  L=000076  BLOCKS  WXX       CM000014
          PROGRAM   VOICON  L=000603  BLOCKS  VOICOM    CM000622
          PROGRAM   ABSMRG  L=001033  BLOCKS            CM011324  MAIN    CM000004  XABS    CM000035  OPANL   CM000004
                                      BLOCKS  NPANL     CM000004
          PROGRAM   EMINIT  L=000414  BLOCKS            CM011636  MAIN    CM000004  NEW     CM000035  OPANL   CM000004
          PROGRAM   EMUP    L=00F254  BLOCKS            CM016076  MAIN    CM000004  XEMISS  CM000035  OPANL   CM000004
                                      BLOCKS  NPANL     CM000004
          PROGRAM   EMDOWN  L=006256  BLOCKS            CM016076  MAIN    CM000004  XEMISS  CM000035  OPANL   CM000004
                                      BLOCKS  NPANL     CM000004
READING FILE        LGO
          PROGRAM   HIRACL  L=002267  BLOCKS            CM014513  MAIN    CM000004  NEW     CM000035  XSUB    CM000012
                                      BLOCKS  WXX       CM000014  SUB1    CM000012  XPANEL  CM000006  XTIME   CM000004
          PROGRAM   CONVFNL L=000232  BLOCKS  NEW       CM000035  XSUB    CM000012  WXX     CM000014  SUB1    CM000012
                                      BLOCKS  XTIME     CM000004
READING FILE        DGO
          PROGRAM   HIRACD  L=001257  BLOCKS            CM012346  MAIN    CM000004  NEW     CM000035  XSUB    CM000012
                                      BLOCKS  WXX       CM000014  SUB1    CM000012  XPANEL  CM000006  XTIME   CM000004
          PROGRAM   CONVFND L=000171  BLOCKS  NEW       CM000035  XSUB    CM000012  WXX     CM000014  SUB1    CM000012
                                      BLOCKS  XTIME     CM000004
          PROGRAM   PANELD  L=000240  BLOCKS  NEW       CM000035  XSUB    CM000012  SUB1    CM000012  XPANEL  CM000006
                                      BLOCKS  XTIME     CM000004
CONSULTING LIBRARY  FORTRAN
          PROGRAM   ACOSIN= L=000071
          PROGRAM   EXP     L=000075
          PROGRAM   SQRT    L=000043
          PROGRAM   ALOG    L=000073
          PROGRAM   OUTC=   L=000172  BLOCKS  FCL.C.    CM000023  Q8.IO.  CM000133
          PROGRAM   FORSYS= L=000503  BLOCKS  Q8.IO.    CM000133  FCL.C.  CM000023  STP.END CM000001
          PROGRAM   UNIT    L=000050  BLOCKS  FCL.C.    CM000023
```

TABLE B.4. LOAD MAP (continued)

```
         PROGRAM    BUFIT=   L=[illegible]
         PROGRAM    CLOCK=   L=000031
         PROGRAM    BUFOUT=  L=000075
         PROGRAM    XTOY=    L=000007
         PROGRAM    XTOI=    L=000010
         PROGRAM    REWIND=  L=000037
         PROGRAM    OUTF=    L=000163  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    INPC=    L=000162  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    08NTRY=  L=000000  BLOCKS  STP,END  CM000001  FCL.C.   CM000023  08.IO.   CM000137
         PROGRAM    ENDFIL=  L=000061
         PROGRAM    GOTOEP=  L=000014
         PROGRAM    SYS=1ST  L=000062
         PROGRAM    FMTAP=   L=000352  BLOCKS  FCL.C.   CM000023  08.IO.   CM000133
         PROGRAM    FECMSK=  L=000041
         PROGRAM    FORUTL=  L=000016  BLOCKS  FCL.C.   CM000023
         PROGRAM    GETFIT=  L=000042
         PROGRAM    BUFIO=   L=000125
         PROGRAM    LDOUT=   L=000241  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    SYSAID=  L=000001
         PROGRAM    COMIO=   L=000364  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    KODER=   L=000456  BLOCKS  FCL.C.   CM000023  08.IO.   CM000133
         PROGRAM    FLTOUT=  L=000311  BLOCKS  08.IO.   CM000133
         PROGRAM    KRAKER=  L=000405  BLOCKS  FCL.C.   CM000023  08.IO.   CM000133
         PROGRAM    OUTCOM=  L=000154  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    INCOM=   L=000276  BLOCKS  08.IO.   CM000133  FCL.C.   CM000023
         PROGRAM    FLTIN=   L=000154  BLOCKS  08.IO.   CM000133
CONSULTING LIBRARY  SYSIO
         PROGRAM    PUT.SO   L=001400  BLOCKS  TERM.RM  CM000001  ADB.RM   CM000010  CON.RM   CM000006  PUT.FO   CM000007
                                       BLOCKS  JMPS.RM  CM000011
         PROGRAM    CLSF.RM  L=000025  BLOCKS  CLSF.FO  CM000007
         PROGRAM    OPEN.SO  L=000254  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  OPEN.FO  CM000007
         PROGRAM    OPEN.RM  L=000277  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  MEMC.RM  CM000003
                                       BLOCKS  OPES.FO  CM000031  OPEN.FO  CM000007
         PROGRAM    Z.SO     L=000101  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  GET.RT   CM000011
         PROGRAM    SKFL.SO  L=000051  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  SKFL.FO  CM000007
         PROGRAM    CHEK.RM  L=000107  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    GTWR.SO  L=000341  BLOCKS  GET.FO   CM000007  TERM.RM  CM000001
         PROGRAM    PTWR.SO  L=000362  BLOCKS  PUT.FO   CM000007  TERM.RM  CM000001  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    REM.SO   L=000333  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  REM.FO   CM000007
         PROGRAM    WLOX.SO  L=000150  BLOCKS  JMPS.RM  CM000011  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    GET.SO   L=001134  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  TERM.RM  CM000001  GET.FO   CM000007
                                       BLOCKS  GET.BT   CM000035  GET.RT   CM000011
         PROGRAM    BTRT.SO  L=000115  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  GET.BT   CM000005
         PROGRAM    OPES.SO  L=000121  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  OPES.FO  CM000001
         PROGRAM    SRF.SO   L=000002
         PROGRAM    CHWR.SO  L=000007  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    CLSV.SO  L=000137  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  CLSV.FO  CM000007
         PROGRAM    ERP.RM   L=000004  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    MOVE.RM  L=000064  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    MCT.RM   L=000233
         PROGRAM    OPEX.SO  L=000014  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    CIO.RM   L=000040  BLOCKS  CON.RM   CM000006
         PROGRAM    CLSF.SO  L=000134  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006  JMPS.RM  CM000011  CLSF.FO  CM000007
         PROGRAM    OSUB.RM  L=000071  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    WAP.SO   L=000260  BLOCKS  ADB.RM   CM000010  CON.RM   CM000006
         PROGRAM    FSU.SO   L=000106  BLOCKS  ADB.RM   CM000010  CON.RM   CM000005  GET.RT   CM000011
         PROGRAM    SKSB.SO  L=000101  BLOCKS  TERM.RM  CM000001  ADB.RM   CM000010  CON.RM   CM000006  SKBL.FO  CM000007
```

TABLE B.4. LOAD MAP (continued)

```
                                                                                       1
         PROGRAM    SKSF.SQ L=000050  BLOCKS  ACO.PM  CM000010  COM.PM  CM000006  SKFL.FO CM000007  SKSF.FO CM000001
         PROGRAM    RLEQ.PM L=000042  BLOCKS  ACO.PM  CM000016  COM.FM  CM000006  PUT.ST  CM000011
CONSULTING LIBRARY  NUCLEUS
         PROGRAM    SYS.PM   L=050037

>>>>>>>>>> SEGMENT  FASCODE     FWA =      +001020 EXEC.FWA=  +101027  LWA+1=  +130504
     ***  BLOCK NAME..TYPE.........FWA.......LENGTH
          FASCODE     PROGRAM      +001327    0012735
          ACOSIN=     PROGRAM      +013334    0000071
          EXP         PROGRAM      +013425    0000075
          SQRT        PROGRAM      +013522    0000047
          ALOG        PROGRAM      +013565    0000073
          OUTC=       PROGRAM      +013660    0000172
          FORSYS=     PROGRAM      +014352    0000603
          UNIT        PROGRAM      +014555    0000050
          BUFIN=      PROGRAM      +014725    0000046
          CLOCK=      PROGRAM      +014773    0000031
          BUFOUT=     PROGRAM      +015024    0000035
          XTOY=       PROGRAM      +015061    0000007
          XTOI=       PROGRAM      +015070    0000010
          REWIND=     PROGRAM      +015100    0000037
          OUTF=       PROGRAM      +015137    0000163
          INPC=       PROGRAM      +015322    0000160
          OBNTRY=     PROGRAM      +015502    0000003
          ENDFIL=     PROGRAM      +015502    0000061
          GOTOER=     PROGRAM      +015563    0000014
          SYS=1ST     PROGRAM      +015577    0000062
          FMTAP=      PROGRAM      +015661    0000352
          FECMSK=     PROGRAM      +016233    0000041
          FORUTL=     PROGRAM      +016274    0000016
          GETFIT=     PROGRAM      +016312    0000042
          BUFIO=      PROGRAM      +016354    0000125
          LDOUT=      PROGRAM      +016501    0000241
          SYSAIO=     PROGRAM      +016742    0000001
          COMIO=      PROGRAM      +016743    0000064
          KODER=      PROGRAM      +017027    0000456
          FLTOUT=     PROGRAM      +017505    0000311
          KRAKER=     PROGRAM      +020016    0000406
          OUTCOM=     PROGRAM      +020424    0000154
          INCOM=      PROGRAM      +020600    0000276
          FLTIN=      PROGRAM      +021076    0000154
          PUT.SQ      PROGRAM      +021252    0001400
          CLSF.PM     PROGRAM      +022652    0000025
          OPEN.SQ     PROGRAM      +022677    0000254
          OPEN.RM     PROGRAM      +023153    0000237
          Z.SQ        PROGRAM      +023412    0000101
          SKFL.SQ     PROGRAM      +023513    0000051
          CHEK.PM     PROGRAM      +023564    0000107
          GTWR.SQ     PROGRAM      +023673    0000041
          PTWR.SQ     PROGRAM      +023734    0000362
          REW.SQ      PROGRAM      +024316    0000033
          WEOR.SQ     PROGRAM      +024351    0000152
          GET.SQ      PROGRAM      +024221    0001134
          STRT.SQ     PROGRAM      +025355    0000115
          OPES.SQ     PROGRAM      +025472    0000121
          SBF.SQ      PROGRAM      +025613    0000002
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TABLE B.4. LOAD MAP (continued)

```
CMM...      PROGRAM     +0.. 1      ....
CLSV.SQ     PROGRAM     +025624     0000137
EMR.RK      PROGRAM     +025763     0000404
MOVE.RM     PROGRAM     +026367     0000064
MCT.RM      PROGRAM     +026453     0000237
OPEX.SQ     PROGRAM     +026786     0000014
CID.PM      PROGRAM     +026722     0000040
CLSF.SQ     PROGRAM     +026762     0000134
DSJR.RM     PROGRAM     +027116     0000071
WAR.SQ      PROGRAM     +027207     0000263
FSU.SQ      PROGRAM     +027467     0000106
SKSB.SQ     PROGRAM     +027575     0000101
SKSF.SQ     PROGRAM     +027676     0000050
RLEO.PM     PROGRAM     +027746     0000042
SYS.PM      PROGRAM     +030010     0000037
PUT.RT      COMMON      +030047     0000011
SKSF.FO     COMMON      +030060     0000001
SKSB.FO     COMMON      +030061     0000001
SKBL.FO     COMMON      +030062     0000007
CLSV.FO     COMMON      +030071     0000007
GET.BT      COMMON      +030100     0000005
REA.FO      COMMON      +030105     0000007
GET.FO      COMMON      +030114     0000007
SKFL.FO     COMMON      +030123     0000007
GET.PT      COMMON      +030132     0000011
OPES.FO     COMMON      +030143     0000001
MEMC.RM     COMMON      +030144     0000003
OPEN.FO     COMMON      +030147     0000007
CLSF.FO     COMMON      +030156     0000007
JMPS.PM     COMMON      +030165     0000011
PUT.FO      COMMON      +030176     0000007
COV.RM      COMMON      +030205     0000006
ABS.RM      COMMON      +030213     0000010
TERM.RM     COMMON      +030223     0000001
STP.END     COMMON      +030224     0000001
OB.IO.      COMMON      +030225     0000133
FCL.C.      COMMON      +030360     0000023
BUF MD      COMMON      +030403     0000040
NEW         GLOBAL      +030443     0000035
MAIN        GLOBAL      +030500     0000004

>>>>>>>>> SEGMENT  HIRACV       FWA =       +030504 EXEC.FWA=   +030504  LWA+1=  +036573
   ***  BLOCK NAME..TYPE........FWA.......LENGTH
    SHEPEL      PROGRAM     +030504     0000266
    SHAFED      PROGRAM     +030772     0000076
    VOICON      PROGRAM     +031070     0000103
    PDFILE      PROGRAM     +031173     0000163
    CONVFMT     PROGRAM     +031256     0000247
    PANEL       PROGRAM     +031525     0000503
    MOLEC       PROGRAM     +032230     0000252
    OVRFAC      PROGRAM     +032502     0000111
    HIRACV      PROGRAM     +032613     0002735
    XMOLEC      COMMON      +035550     0000124
    XPANEL      COMMON      +035674     0000005
    XTIME       COMMON      +035702     0000004
    SUB1        COMMON      +035706     0000012
    BUF2        COMMON      +035720     0000007
```

TABLE B.4. LOAD MAP (continued)

```
         XSUB      COMMON      [illegible] [illegible]
         VOICOM    COMMON      +035735   0000622
         WXX       COMMON      +036557   0000014
     *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         NEW       FASCODE     SAFE      +030443
         MAIN      FASCODE     SAFE      +030599

>>>>>>>>> SEGMENT  HIRACL      FWA =     +030504 EXEC.FWA=  +030504  LWA+1=  +035165
     ***  BLOCK NAME..TYPE........FWA......LENGTH
         SHAPEL    PROGRAM     +030534   0000266
         ROFILF    PROGRAM     +030772   0000163
         CONVFNL   PROGRAM     +031155   0000232
         PANEL     PROGRAM     +031407   0000503
         MOLEC     PROGRAM     +032112   0000252
         OVRFAC    PROGRAM     +032364   0000111
         HIRACL    PROGRAM     +032475   0002267
         XMOLEC    COMMON      +034764   0000124
         XPANEL    COMMON      +035110   0000006
         XTIME     COMMON      +035116   0000004
         SUB1      COMMON      +035122   0000012
         BUF2      COMMON      +035134   0000003
         XSUE      COMMON      +035137   0000012
         WXX       COMMON      +035151   0000014
     *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         NEW       FASCODE     SAFE      +030443
         MAIN      FASCODE     SAFE      +030599

>>>>>>>>> SEGMENT  HIRACD      FWA =     +030504 EXEC.FWA=  +030504  LWA+1=  +033461
     ***  BLOCK NAME..TYPE........FWA......LENGTH
         SHAPED    PROGRAM     +030504   0000076
         ROFILF    PROGRAM     +030572   0000153
         CONVFND   PROGRAM     +030765   0000171
         PANELD    PROGRAM     +031156   0000240
         MOLEC     PROGRAM     +031416   0000252
         OVRFAC    PROGRAM     +031670   0000111
         HIRACD    PROGRAM     +032001   0001257
         XMOLEC    COMMON      +033260   0000124
         XPANEL    COMMON      +033404   0000006
         XTIME     COMMON      +033412   0000004
         SUB1      COMMON      +033416   0000012
         BUF2      COMMON      +033430   0000003
         XSUB      COMMON      +033433   0000012
         WXX       COMMON      +033445   0000014
     *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         NEW       FASCODE     SAFE      +030443
         MAIN      FASCODE     SAFE      +030500

>>>>>>>>> SEGMENT  ABSHPG      FWA =     +030504 EXEC.FWA=  +030504  LWA+1=  +031604
     ***  BLOCK NAME..TYPE........FWA......LENGTH
         ABSHPG    PROGRAM     +030504   0001023
         NPANL     COMMON      +031537   0000004
         OPANL     COMMON      +031543   0000004
         XABS      COMMON      +031547   0000035
     *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         MAIN      FASCODE     SAFE      +030500
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDQ

TABLE B.4. LOAD MAP (continued)

```
>>>>>>>>>> SEGMENT  EMINIT      FWA =      +030504 EXEC.FWA=   +030504  LWA+1=  +031124
    ***  BLOCK NAME..TYPE.........FWA.......LENGTH
         EMINIT      PROGRAM     +030504   0000414
         OPANL       COMMON      +031120   0000004
    *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         NEW         FASCODE     SAFE      +030443
         MAIN        FASCODE     SAFE      +030510

>>>>>>>>>> SEGMENT  EMUP        FWA =      +030504 EXEC.FWA=   +030504  LWA+1=  +037025
    ***  BLOCK NAME..TYPE.........FWA.......LENGTH
         EMUP        PROGRAM     +030504   0006254
         NPANL       COMMON      +036760   0000004
         OPANL       COMMON      +036764   0000004
         XEMISS      COMMON      +036770   0000035
    *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         MAIN        FASCODE     SAFE      +030500

>>>>>>>>>> SEGMENT  EMDOWN      FWA =      +030504 EXEC.FWA=   +030504  LWA+1=  +037027
    ***  BLOCK NAME..TYPE.........FWA.......LENGTH
         EMDOWN      PROGRAM     +030504   0006256
         NPANL       COMMON      +036762   0000004
         OPANL       COMMON      +036766   0000004
         XEMISS      COMMON      +036772   0000035
    *** FN.GLOBAL..SEGMENT...SAFETY......FWA
         MAIN        FASCODE     SAFE      +030500
******** CM BLANK COMMON FWA=  +037027  LWA+1=  0155125
```

TABLE B.5. TEST CASE OUTPUT

```
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER              EMISS UP
   1    1    0    0    0

   SECANT =        1.00000000
IHIRAC = 1
 V1 , V2  2095. 2105.
 TBOUND   273.

1  LINE FILE FOR THE FASCODE REPORT                                        03/29/79  10.24.35.

      H20  = 1
      CO2  = 1
       O3  = 1
      N2O  = 1
       CO  = 1
      CH4  = 1
       O2  = 1

 LOWEST LINE =   1990.000  HIGHEST LINE =   2210.000


NLAYER 1
FIRST DV = .01375082 7
.0138 -1


      VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER              EMISS UP

 OUTPUT FILE NO. =   10

 SECANT =         1.00000

 PRESS(MB) =    525.85660

 TEMP(K) =      264.38

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =   9.380E+21
      CO2  =   1.469E+21
       O3  =   2.611E+17
      N2O  =   1.195E+18
       CO  =   3.201E+17
      CH4  =   6.831E+18
       O2  =   8.942E+23

DV(CM-1) =      .01380080

V1(CM-1) =     2095.0000

V2(CM-1) =     2105.0000

BOUNDVS(CM-1) =  3.5328

        TIME           READ    CONVOLUTION        PANEL
        5.952          .314           .703         .005
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
 AVERAGE WIDTH =      .84914   AVERAGE ZETA =      .96917   NO. LINES =        957   NO. CHANGES =
 EMINIT, WFILE, KFILE  11 10
.04399999999990 SECS WERE NEEDED FOR EMINIT

 NLAYER 2
FIRST DV = .J89097795134351
COMPUTED DV BEFORE MODIFICATION = .C09097705134351
.0092 ?


      VOIGT
1  APPROX TO KYLF REPORT....MIDLATITUDE SUMMER              EMISS UF

 OUTPUT FILE NO. =   10

 SECANT =         1.00000

 PRESS(MB) =    332.87410

 TEMP(K) =      242.75

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  1.649E+21
      CO2  =  1.284E+21
       O3  =  4.537E+17
      N2O  =  1.869E+18
       CO  =  2.917E+17
      CH4  =  6.224E+18
       O2  =  8.149E+23

 DV(CM-1) =     .00920600

 V1(CM-1) =     2095.0000

 V2(CM-1) =     2135.0030

 BCUNDVS(CM-1) =  2.3552

          TIME            READ    CONVOLUTION           PANEL
         6.835            .026           .616            .611

 AVERAGE WIDTH =     .03256   AVERAGE ZETA =     .94373   NO. LINES =        926   NO. CHANGES =          0
 EMUP, MFILE, LFILE, KFILE  12 11 10

 THE TIME AT THE START OF EMUP IS        6.842
 THIS OUTPUT IS WRITTEN ON FILE  12
 P= .6666666666667
 P= .3333333333333
      1 2095.000000  .12745475E-07  .86547116E+03
   1000 2105.903400  .16022095E-08  .98387814E+30

 THE TIME AT THE END OF EMUP IS        6.954
      .112 SECS WERE REQUIRED FOR THIS ADDITION
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
 NLAYER 3
FIRST DV = .005657652914983
COMPUTED DV BEFORE MODIFICATION = .005657652914983
.006133333333333 2


      VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER                EMISS UP

 OUTPUT FILE NO. =   10

 SECANT =        1.00000

 PRESS(MB) =    197.56440

 TEMP(K) =      222.87

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  6.815E+19
      CO2  =  6.262E+20
       O3  =  5.351E+17
      N2O  =  5.313E+17
       CO  =  1.423E+17
      CH4  =  3.035E+18
       O2  =  3.975E+23

 DV(CM-1) =    .00613333

 V1(CM-1) =    2095.0000

 V2(CM-1) =    2105.0000

 BOUNDVS(CM-1) =  1.5701

        TIME            READ    CONVOLUTION          PANEL
       7.633            .026           .493           .030

 AVERAGE WIDTH =     .02028   AVERAGE ZETA =     .91514   NO. LINES =       701    NO. CHANGES =
 EMUP, MFILE, LFILE, KFILE  11 12 10

 THE TIME AT THE START OF EMUP IS        7.638
 THIS OUTPUT IS WRITTEN ON FILE  11
 P= .6666666666667
 P= .3333333333333
         1 2095.000000  .13365235E-07  .82352182E+00
      1405 2103.629600  .32450155E-08  .95188694E+00
         1 2103.535733  .30887121E-08  .95452910E+00
       223 2104.097333  .17457414E-08  .97130757E+00

 THE TIME AT THE END OF EMUP IS        7.791
       .153 SECS WERE REQUIRED FOR THIS ADDITION
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TABLE B.5. TEST CASE OUTPUT (continued)

```
NLAYER 4
FIRST DV = .00364065265806I
COMPUTED DV BEFORE MODIFICATION = .00364065265080I
.00336666666667 1


      VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER                EMISS UP

 OUTPUT FILE NO. =    13

 SECANT =          1.00000

 PRESS(MB) =    124.38000

 TEMP(K) =      216.00

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  7.612E+18
      CO2  =  4.043E+20
       O3  =  7.852E+17
      N2O  =  3.429E+17
       CO  =  9.185E+16
      CH4  =  1.960E+18
       O2  =  2.566E+23

 DV(CM-1) =     .00386667

 V1(CM-1) =    2095.0000

 V2(CM-1) =    2105.0300

 ROUNDVS(CM-1) =    .7851

          TIME            READ    CONVOLUTION          PANEL
         8.449            .017          .480            .044

 AVERAGE WIDTH =     .01311   AVERAGE ZETA =     .87572   NO. LINES =        554    NO. CHANGES =         4
 EMUP, MFILE, LFILE, KFILE  12 11 10

 THE TIME AT THE START OF EMUP IS        8.456
 THIS OUTPUT IS WRITTEN ON FILE  12
 P= .5
       1 2095.000000  .14056810E-07  .75072151E+00
    1488 2099.314000  .76514847E-08  .89597090E+00
       1 2099.317867  .73994920E-08  .89964745E+00
    1654 2105.060400  .19737855E-08  .95772057E+00

 THE TIME AT THE END OF EMUP IS        8.739
      .283 SECS WERE REQUIRED FOR THIS ADDITION
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
NLAYER 5
FIRST DV = .002319662190185
COMPUTED DV BEFORE MODIFICATION = .002319662190185
.5823 3


    VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER              EMISS UP

 OUTPUT FILE NO. =   10

 SECANT =        1.00000

 PRESS(MB) =     77.27710

 TEMP(K) =     216.92

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  5.845E+18
      CO2  =  2.539E+20
       O3  =  1.081E+18
      N2O  =  2.129E+17
       CO  =  5.732E+16
      CH4  =  1.216E+18
       O2  =  1.593E+23

 DV(CM-1) =     .00230000

 V1(CM-1) =     2095.0000

 V2(CM-1) =     2105.0090

 BOUNDV5(CM-1) =    .5890

        TIME           READ    CONVOLUTION          PANEL
       9.318           .020           .388           .061

 AVERAGE WIDTH =     .30843   AVERAGE ZETA =     .81718   NO. LINES =        511    NO. CHANGES =
 EMUP, AFILE, LFILE, KFILE  11 12 10

 THE TIME AT THE START OF EMUP IS          9.327
 THIS OUTPUT IS WRITTEN ON FILE  11
 P= .75
  = .5
 P= .25
       1 2095.000006  .15354102E-07  .62425535E+00
    1400 2098.235106  .33336669E-08  .86539437E+00
       1 2098.230400  .33239693E-08  .86914675E+00
    2400 2103.755100  .45419312E-08  .78461920E+00
       1 2103.750400  .47629036E-08  .76232018E+00
     541 2105.000400  .20418805E-08  .94927946E+00

 THE TIME AT THE END OF EMUP IS          9.734
      .407 SECS WERE REQUIRED FOR THIS ADDITION
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
NLAYER 6
FIRST DV = .001219069804383
COMPUTED DV BEFORE MODIFICATION = .0012190690064387
.00115 1


      VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER                    EMISS UP

 OUTPUT FILE NO. =    13

 SECANT =          1.00000

 PRESS(MB) =     36.63338

 TEMP(K) =      223.89

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  1.349E+19
      CO2  =  3.264E+20
       O3  =  3.311E+19
      N2O  =  2.786E+17
       CO  =  7.462E+16
      CH4  =  1.592E+18
       O2  =  2.805E+23

 DV(CM-1) =     .00115000

 V1(CM-1) =     2095.6003

 V2(CM-1) =     2105.0002

 BOUNDVS(CM-1) =    .2944

          TIME             READ     CONVOLUTION            PANEL
         10.367            ..16           .383              .131

  AVERAGE WIDTH =      .08459   AVERAGE ZETA =     .67534   NO. LINES =        457    NO. CHANGES =
 EMUP, MFILE, LFILE, KFILE  12 11 10

 THE TIME AT THE START OF EMUP IS        10.775
 THIS OUTPUT IS WRITTEN ON FILE  12
 P= .5
          1 2095.000000  .21756529E-07  .21564547E+03
       1403 2095.616056  .37772926E-08  .69457932E+03
          1 2096.619280  .37756749E-08  .89418792E+03
       2466 2099.374053  .12613086E-07  .36932559E+03
          1 2099.379283  .13356822E-07  .34363177E+03
       2430 2102.139056  .19375333E-08  .94436763E+03
          1 2102.139296  .19477931E-08  .94374616E+03
       2430 2104.893555  .56438156E-08  .66539722E+03
          1 2104.899230  .51447629E-08  .71734273E+03
         89 2105.000400  .25326913E-06  .96578529E+03

 THE TIME AT THE END OF EMUP IS        11.083
      .708 SECS WERE REQUIRED FOR THIS ADDITION
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
 NLE EP 7
FIRST DV = .00061514227500 1
COMPUTED DV BEFORE MODIFICATION = .00061514227500 1
.000575 1


      VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER              FRISS UF

 OUTPUT FILE NO. =   10

 SECANT =         1.00000

 PRESS(MB) =      0.34020

 TEMP(K) =     242.96

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  4.989E+18
      CO2  =  7.053E+19
       O3  =  1.259E+18
      N2O  =  5.982E+16
       CO  =  1.603E+16
      CH4  =  3.419E+17
       O2  =  4.477E+22

 DV(CM-1) =     .00057500

 V1(CM-1) =     2095.0030

 V2(CM-1) =     2105.0000

 BOUNDVS(CM-1) =    .1472

        TIME            READ    CONVOLUTION          PANEL
       11.012           .018           .360           .247

 AVERAGE WIDTH =    .03229   AVERAGE ZETA =    .31237   NO. LINES =       476    NO. CHANGES =        0
 EMUP, MFILE, LFILE, KFILE  11 12 10

 THE TIME AT THE START OF EMUP IS :      11.018
 THIS OUTPUT IS WRITTEN ON FILE  11
 P= .5
      1 2095.009000  .27902461E-07  .00214101E-01
   1400 2095.009025  .21160066E-08  .96931011E+00
      1 2095.039600  .21152274E-08  .96992127E+00
   2400 2097.189025  .14366695E-07  .83259670E+00
      1 2097.189600  .14461086E-07  .83164412E+00
   2400 2098.569025  .44750552E-08  .97383536E+00
      1 2098.569600  .44948103E-08  .93334341E+00
   2400 2099.949025  .13551065E-07  .43647145E+00
      1 2099.949600  .12996471E-07  .46983639E+01
   2400 2101.329025  .37595367E-06  .92484192E+00
      1 2101.329600  .37512967E-08  .92402[illegible]E+00
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
    2400 2102.709025  .23876393E-08  .95595296E+00
       1 2102.709600  .23126826E-08  .95669042E+00
    2403 2104.009025  .28996746E-08  .90195952E+00
       1 2104.009600  .28834147E-08  .90712399E+00
    1504 2104.999025  .27381791E-08  .90846186E+00

 THE TIME AT THE END OF EMUP IS       13.196
      1.378 SECS WERE REQUIRED FOR THIS ADDITION


 NLAYER 4
FIRST DV = .0005429977640457
COMPUTED DV BEFORE MODIFICATION = ..0005429977640457
.000575 0


       VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER              EMISS UP

 OUTPUT FILE NO. =   10

 SECANT =        1.00000

 PRESS(MB) =      2.55890

 TEMP(K) =      263.20,

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  4.913E+17
      CO2  =  1.113E+19
      O3   =  1.529E+17
      N2O  =  9.484E+15
      CO   =  2.541E+15
      CH4  =  5.428E+16
      O2   =  7.697E+21

 DV(CM-1) =     .00057500

 V1(CM-1) =     2095.0000

 V2(CM-1) =     2105.0000

 BOUNDVS(CM-1) =    .1472

          TIME              READ    CONVOLUTION           PANEL
        13.875              .027           .310            .235

  AVERAGE WIDTH =      .00262   AVERAGE ZETA =      .11532   NO. LINES =         437    NO. CHANGES =          3
 EMUP, MFILE, LFILE, KFILE  12 11 10

 THE TIME AT THE START OF EMUP IS        13.878
 THIS OUTPUT IS WRITTEN ON FILE  12
          1 2095.000000  .29255767E-07  .65129594F-01
       1468 2095.839025  .21169315F-08  .96981773E+00
          1 2095.839600  .21152780E-08  .96902175E+00
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
2400 2097.109825  .14371050E-07  .82256112E+00
   1 2097.109838  .14465357E-07  .83100742E+00
2400 2098.569825  .44770831E-08  .97301706E+00
   1 2098.569630  .44964740E-08  .83332556E+00
2400 2099.949825  .13654558E-07  .43556150E+00
   1 2099.949688  .13080087E-07  .46985685E+00
2400 2101.329825  .37553172E-08  .92486621E+00
   1 2101.329600  .37582693E-08  .92397671E+00
2400 2102.709825  .23393096E-08  .95693650E+00
   1 2102.709600  .23136859E-08  .95687406E+00
2400 2104.089825  .29252031E-08  .96173432E+00
   1 2104.089600  .20256766E-08  .96692761E+00
1504 2104.999825  .27471985E-08  .90025120E+00

THE TIME AT THE END OF EMUP IS          14.696
     .018 SECS WERE REQUIRED FOR THIS ADDITION


NLAYER 9
FIRST DV = .00052758307980403
COMPUTED DV BEFORE MODIFICATION = .00052758307980403
.000575 8


       VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER                   EMISS UP

 OUTPUT FILE NO. =    10

 SECANT =         1.00000

 PRESS(MB) =      1.15590

 TEMP(K) =      267.25

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  2.598E+17
      CO2  =  8.437E+18
       O3  =  6.597E+16
      N2O  =  7.156E+15
       CO  =  1.917E+15
      CH4  =  4.890E+16
       O2  =  5.355E+21

 DV(CM-1) =     .00057590

 V1(CM-1) =    2095.0000

 V2(CM-1) =    2105.0000

 BOUNDVS(CM-1) =   .1472

        TIME            READ     CONVOLUTION          PANEL
       15.361           .017            .312           .233
```

TABLE B.6. TEST CASE OUTPUT (continued)

```
 AVERAGE WIDTH =      .08196   AVERAGE ZETA =     .05524   NO. LINES =        43.     NO. CHANGES =          8
EHUP, NFILE, LFILE, KFILE  11 12 10

THE TIME AT THE START OF EHUP IS        15.365
THIS OUTPUT IS WRITTEN ON FILE  11
        1 2095.006000  .29061931E-07  .64565708E-01
     1400 2095.009025  .21169594E-08  .96981753E+00
        1 2095.009600  .21153046E-08  .96982857E+00
     2400 2097.109025  .14371696E-07  .83255564E+00
        1 2097.109600  .14466224E-07  .83188115E+00
     2400 2098.569025  .44783712E-08  .93301431E+00
        1 2098.569600  .44965009E-08  .93332191E+00
     2400 2099.949025  .13670985E-07  .43540242E+00
        1 2099.949600  .13106583E-07  .46050485E+00
     2400 2101.329025  .37564435E-08  .92479193E+00
        1 2101.329600  .37594515E-08  .92396795E+00
     2400 2102.769025  .23097464E-08  .95693327E+00
        1 2102.709600  .23143293E-08  .95687681E+00
     2400 2104.049025  .29312077E-08  .96160967E+00
        1 2104.109600  .28308967E-08  .96648076E+00
     1504 2104.999025  .27511759E-08  .36822146E+00

THE TIME AT THE END OF EHUP IS       16.186
      .821 SECS WERE REQUIRED FOR THIS ADDITION

                              ----  -  -   --------  ----------------

 NLAYER 10
FIRST DV = .0385845393216063
COMPUTED DV BEFORE MODIFICATION = .0005845393216063
.600575 0


       VOIGT
1  APPROX TO KYLE REPORT....MIDLATITUDE SUMMER    LAST LAYER EMISS UP

 OUTPUT FILE NO. =   10

 SECANT =        1.30600

 PRESS(MB) =      .27420

 TEMP(K) =     255.42

 COLUMN DENSITY(MOLECULES/CM**2)

      H2O  =  4.749E+16
      CO2  =  4.380E+16
       O3  =  1.176E+16
      N2O  =  3.464E+15
       CO  =  9.276E+14
      CH4  =  1.579E+16
       O2  =  2.591E+21

 DV(CM-1) =    .0005750A
```

TABLE B.5. TEST CASE OUTPUT (continued)

```
V1(CM-1) =    2095.0000

V2(CM-1) =    2105.0000

BOUNDVS(CM-1) =   .1472

         TIME              READ    CONVOLUTION          FANEL
       16.036              .017          .307            .228

 AVERAGE WIDTH =      .30107   AVERAGE ZETA =     .01435   NO. LINES =        430    NO. CHANGES =          9
EMUP, MFILE, LFILE, KFILE  12 11 10

THE TIME AT THE START OF EMUP IS        16.839
THIS OUTPUT IS WRITTEN ON FILE  12
       1 2095.000000  .29914361E-07  .63689568E-01
    1408 2095.809025  .21169607E-08  .96901752E+00
       1 2095.809600  .21153866E-08  .96902655E+00
    2400 2097.189025  .14371715E-07  .83255475E+00
       1 2097.189600  .14466249E-07  .83100684E+00
    2400 2098.569025  .44782849E-08  .93381414E+00
       1 2098.569600  .44965956E-08  .93332173E+00
    2400 2099.949025  .13675607E-07  .43539277E+00
       1 2099.949600  .13107186E-07  .46990136E+00
    2400 2101.329025  .37564752E-08  .52479157E+00
       1 2101.329600  .37594050E-08  .92396753E+00
    2400 2102.709025  .23897556E-06  .95693316E+00
       1 2102.709600  .23143379E-08  .95687070E+00
    2400 2104.089025  .29313654E-08  .90168769E+00
       1 2104.089600  .28310341E-08  .90688706E+00
    1504 2104.999025  .27512833E-08  .96022110E+00

THE TIME AT THE END OF EMUF IS       17.654
     .915 SECS WERE REQUIRED FOR THIS ADDITION
```

USAGE NOTES

1. The user must structure the input atmosphere such that the ratio of the previous DV to the present DV is acceptable to FASCODE, (see the sample input data).

2. The program is written for execution on a CDC 6600 computer system at Hanscom AFB. Accordingly, the control cards input/output statements and plot options need to be tailored to the user's installation.

APPE[illegible]

PROGRAM LISTING

LISTING OF FASCODE PROGRAM

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      PROGRAM FASCODE(INPUT=130/1300,OUTPUT=130/1300,TAPE5=INPUT
     C ,TAPE6=OUTPUT,TAPE7,TAPE10,TAPE11,TAPE12)
C
C FAST SIGNATURE CODE
C
      COMMON COMSTR(250,9)
      COMMON FF(3600),SF(900),VSF(225)
      COMMON/MAIN/ KFILE,KPANEL,P0,TEMP0
      COMMON/NEW/      NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
     C                 DV,V1,V2,NLAYER
      COMMON /BUFHD/ FILHD(10),BMOLID(10),MOLIND(10),VLINLO,VLINHI
C
      NMOL=7
      ITYPE=-1
      P0=1013.
      TEMP0=296.
      KPANEL=0
      LINFIL=9
      MFILE=12
      LFILE=11
      KFILE=10
      NLAYER=0
      PRINT 900
      READ 910,(XID(I),I=1,7)
C
C     IHIRAC = 1 CALL HIRACV
C     IHIRAC = 2 CALL HIRACL
C     IHIRAC = 3 CALL HIRACD
C
C IF IEMIT .EQ. 1  PROGRAM WILL COMPUTE EMISSION
C
C ILOOK=1   LOOKING FROM SPACE TO GROUND FOR EMISSION
C ILOOK=0   LOOKING FROM GROUND TO SPACE FOR EMISSION
C
      READ 920, IHIRAC,IEMIT,ILOOK,ISCAN,IPLOT
      IF (IEMIT.EQ.0) XID(7)=10H ABS COEFF
      IF ( (IEMIT.EQ.1 ) .AND. (ILOOK.EQ.0) ) XID(7)=10H EMISS UP
      IF ( (IEMIT.EQ.1 ) .AND. (ILOOK.EQ.1) ) XID(7)=10H EMISS DWN
      PRINT 930, XID
       PRINT 920, IHIRAC,IEMIT,ILOOK,ISCAN,IPLOT
      READ 940,SECANT
      PRINT 950,SECANT
      PRINT*,"IHIRAC = ",IHIRAC
C
C TBOUND IS A TEMPERATURE WHICH WILL BE USED TO COMPUTE A
C BOUNDRY.  WHEN TBOUND = 0, NO BOUNDRY WILL BE USED.
C
      READ 940, V1,V2
      PRINT *, " V1 , V2  " , V1,V2
      READ 940, TBOUND
      PRINT *, " TBOUND   " , TBOUND
C
      READ 920,NNLAYR
      BUFFER IN(LINFIL,1) (FILHD(1),VLINHI)
      IF(UNIT(LINFIL).EQ.0) STOP
      PRINT 930, FILHD
      PRINT 905
      PRINT 960, (BMOLID(I),MOLIND(I),I=1,NMOL)
      PRINT 970, VLINLO,VLINHI
      IF(2*(NNLAYR/2).NE. NNLAYR) GO TO 110
      MSTOR=MFILE
      MFILE=LFILE
      LFILE=MSTOR
  110 CONTINUE
      IF (IEMIT.EQ.0) KFILE=MFILE
      DO 280 K=1,NNLAYR
      PRINT 900
      ISMALL=0
      REWIND LINFIL
      BUFFER IN (LINFIL,1) (FILHD(1),VLINHI)
      IF (UNIT(LINFIL) .EQ. 0 ) STOP
      NLAYER=NLAYER+1
      PRINT *, " NLAYER " , NLAYER
      IF (K.EQ.NNLAYR) XID(6)=10HLAST LAYER
```

```
      READ 940,PAVE,TAVE
      READ 948,(WK(IK),IK=1,7)
  120 OLDDV=DV
      DENSCOR=PAVE/P0*SQRT(TEMP0/TAVE)
      ALBAR=(0.10)*DENSCOR
      AVMASS=36.
      ADBAR=3.58178E-07*(0.5*(V1+V2))*SQRT(TAVE/AVMASS)
      AVBAR=0.5*( ALBAR + SQRT ( ALBAR*ALBAR + 4.*ADBAR*ADBAR ) )
      IF(INIRAC.EQ.1) DV=AVBAR/4.
      IF(INIRAC.EQ.2) DV=ALBAR/4.
      IF(INIRAC.EQ.3) DV=ADBAR/4.
  125 CONTINUE
      PRINT*,"FIRST DV = ",DV
      IF (NLAYER.NE.1) GO TO 130
C       DV IS ASSUMED TO BE .LT. 1
      ISCAL=ALOG10(DV) - 3.
      SCAL=10.**ISCAL
      IDV=(DV/SCAL) + 0.5
      DV=SCAL*FLOAT(IDV)
      GO TO 170
 130  TYPE=OLDDV/DV
      IF(TYPE.GE.1.) GO TO 133
      TYPE=1.0/TYPE
      ISMALL=1
  133 PRINT*,"COMPUTED DV BEFORE MODIFICATION = ",DV
C
      IF(TYPE .GT. 2.5) GO TO 150
      IF(TYPE .GE. 1.201) GO TO 138
      DV=OLDDV
      ITYPE=0
      GO TO 170
 138  ITYPE=1./(TYPE-1.0)+0.5
      IF(ISMALL .EQ. 0) DV=OLDDV*FLOAT(ITYPE)/FLOAT(ITYPE+1)
      IF(ISMALL .EQ. 1) DV=OLDDV*FLOAT(ITYPE+1)/FLOAT(ITYPE)
      GO TO 170
 150  CONTINUE
      PRINT 900
      STOP
 170  PRINT *,DV,ITYPE
C
      IF(INIRAC.EQ.1) CALL HIRACV
      IF(INIRAC.EQ.2) CALL HIRACL
      IF(INIRAC.EQ.3) CALL HIRACD
C
                          NPTS=1
      IF (IEMIT.NE.0) GO TO 190
      IF (NLAYER.GT.1) GO TO 180
      GO TO 210
 180  CALL ABSMRG (ITYPE,ISMALL,IOLDPL,LFILE,MFILE)
      GO TO 210
 190  CONTINUE
C     IEMIT=1 TO REACH THIS STATEMENT
      IF (ISMALL.NE.0) STOP " ISMALL .NE. 0 "
      IF (NLAYER.GT.1) GO TO 200
      CALL EMINIT(ILOOK,TBOUND,MFILE)
      GO TO 210
C
 200  IF (ILOOK.EQ.0) CALL EMUP   (NPTS,ITYPE,LFILE,MFILE)
      IF (ILOOK.EQ.1) CALL EMDOWN (NPTS,ITYPE,LFILE,MFILE)
C
 210  CONTINUE
      MSTOR=MFILE
      MFILE=LFILE
      LFILE=MSTOR
 230  REWIND 11
      REWIND 12
      IOLDPL=NPANEL
      NPANEL=0
      KFILE=19
  200 CONTINUE
 330  REWIND 11
C
C
 335  IF (IPLOT.NE. 0 )  CALL TPLOT(IPLOT,MFILE)
 330  STOP
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      C
      C
      C
      900   FORMAT(1H1)
      905   FORMAT(1H0)
      910   FORMAT(10A10)
        920 FORMAT(16I5)
        930 FORMAT(1H0,10A10)
      940   FORMAT(10F10.3)
        950 FORMAT('0   SECANT = ',F15.8 )
      960   FORMAT(5X,A6,3H = ,I1)
      970   FORMAT('0 LOWEST LINE = 'F10.3,'  HIGHEST LINE = ' F10.3)
      980   FORMAT(' THE RATIO OF OLD DV TO NEW DV EXCEEDS 2.0   ' )
      C
            END
```



```
            SUBROUTINE HIRACV
      C
      C
      ***************************************************************************
      *                                                                         *
      *                                                                         *
      *     CALCULATES MONOCHROMATIC ABSORPTION COEFFICIENT FOR SINGLE LAYER   *
      *                                                                         *
      *                                                                         *
      *             USES APPROXIMATE VOIGT ALGORITHM                            *
      *                                                                         *
      ***************************************************************************
      C
      C
            COMMON GNU(250),S(250),ALFA0(250),EPP(250),MOL(250),
           C       EPOPTH(250),RECALF(250),ZETAI(250),IZETA(250)
            COMMON FF(3600),SF(900),VSF(225)
            COMMON/MAIN/ KFILE,KPANEL,P0,TEMP0
            COMMON/NEW/      NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
           C             DV,V1,V2,NLAYER
            COMMON /XSUB/LIMIN,ILC,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
            COMMON /WXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
           1AX
            COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
            COMMON /XPANEL/ V1P,V2P,DVP,NLIM,NSHIFT,NPTS
            COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
            COMMON/VOICOM/AVRAT(201),CLD(201)
            DIMENSICN ALFCOR(7),ALFD1(7),U(7),SCOR(7)
            DIMENSION XF(251),XS(251),XVS(251),XD(251)
      C
            DATA NWF / 4/, DXF / 0.02/, NF /201/, NFMAX /251/
            DATA NWS /16/, DXS / 0.08/, NS /201/, NSMAX /251/
            DATA NWVS/64/, DXVS/ 0.32/, NVS/201/, NVSMAX/251/
      C
            DATA IENTER/0/, LIMIN/250/, NSHIFT/32/, NLIMF/2401/, NPTS/    0/
      C
            DATA SUBID /10H  VOIGT    /
      C
            IF (IENTER.NE.0) GO TO 10
            IENTER=1
            NLIMS=(NLIMF/4)+1
            NLIMVS=(NLIMS/4)+1
      C     NOTE (DXVS/DXF) IS 16 AND (DXVS/DXS) IS 4
            NBOUND=FLOAT(NWVS)*(DXVS/DXF)
            MAXF=NLIMF+NBOUND
            MAXS=(MAXF/4)+1
            MAXVS=(MAXS/4)+1
            CALL VOICON
            CALL SHAPEL(XF,XS,XVS)
            CALL SHAPED(XD)
            CALL MOLEC(1,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
      10    PRINT 900
```

```
      PRINT 970, SUBID
      TIMRDP=TIMCNV=TIMPNL=0.
      PRINT 904,    (XID(I),I=1,7)
      REWIND KFILE
      IEOF=0
      PRINT 910, KFILE
      PRINT 915, SECANT
      PRINT 925, PAVE,TAVE
      PRINT 930, (HMOLID(M),WK(M),M=1,NMOL)
      PRINT 940, DV,V1,V2
      DVP=DV
      DVS=(DXS/DXF)*DV
      DVVS=(DXVS/DXF)*DV
      BOUND=FLOAT(NBOUND)*DV/2.
      BOUNVS=BOUND/2.
      PRINT 942, BOUNVS
      ALFMAX=BOUND/FLOAT(NMVS)
      NLO=NBOUND+
      NHI=NLIMF+NSHIFT-1
      DO 50 I=1,MAXF
50    FF(I)=0.
      DO 60 I=1,MAXS
60    SF(I)=0.
      DO 70 I=1,MAXVS
70    VSF(I)=0.
C
$     WRITE (KFILE) (XID(I),I=1,7),SECANT,PAVE,TAVE,(HMOLID(M),M=1,7),
$    *  (WK(I),I=1,NMOL),DV,V1,V2
      BUFFER OUT(KFILE,1) (XID(1),NLAYER)
      IF (UNIT(KFILE).EQ.0) STOP "99"
C
      VFT=V1-2.*BOUND
      VBOT=V1-BOUND
      VTOP=V2+BOUND
C
      XKT0=0.6951*TEMP0
      XKT =0.6951*TAVE
      XKTFAC=(1./XKT) - (1./XKT0)
      CALL MOLEC(2,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
      DO 80 M=1,NMOL
80    U(M)=WK(M)*SCOR(M)*SECANT
      ICAT=0
      SUMALF=0.
      SUMZET=0.
      NCHNG=0
C
90    CONTINUE
C
100   CONTINUE
C
      CALL SECOND (TIME0)
      IF (IEOF.NE.0) GO TO 150
      CALL RDFILE(GNU,S,ALFA0,EPP,MOL)
      CALL SECOND (TIME)
      TIMRDF=TIMRDF+TIME-TIME0
C
      IF(IEOF.NE.0) GO TO 140
C
C     MODIFY LINE DATA FOR TEMPERATURE,PRESSURE, AND COLUMN DENSITY
C
      DO 130 I=ILO,IHI
      M=MOL(I)
      EFDPTH(I)=S(I)*U(M)
      IF (EFDPTH(I).LE.0.) GO TO 130
      ICNT=ICNT+1
      ALFL=ALFA0(I)*ALFCOR(M)
      ALFAD=GNU(I)*ALFD1(M)
      ZETA=ALFL/(ALFL+ALFAD)
      IZ=100.*ZETA+1.5
      ALFV=AVRAT(IZ)*ALFL
      IF(IZ.LT.101) ALFV=AVRAT(IZ)*ALFAD
      IF(ALFV.GE.DV) GO TO 110
      PRINT 945, GNU(I),S(I),ALFA0(I),ALFV,DV,M
      ALFV=DV
```

```
      NCHNG=NCHNG+1
110   IF (ALFV.LE.ALFMAX) GO TO 120
      PRINT 930, GNU(I),S(I),ALFAR(I),ALFV,ALFMAX,K
      ALFV=ALFMAX
      NCHNG=NCHNG+1
120   CONTINUE
      SUMALF=SUMALF+ALFV
      SUMZET=SUMZET+ZETA
      RECALF(I)=1./ALFV
      EFDPTH(I)=EFDPTH(I)*EXP(-EPP(I)*XKTFAC)*RECALF(I)*
     $          (1.-EXP(-GNU(I)/XKT))/(1.-EXP(-GNU(I)/XKTV))
      ZETAI(I)=ZETA
      IZETA(I)=IZ
130   CONTINUE
      IF (NCHNG.GT.100) GO TO 160
140   CONTINUE
C
      CALL CONVFNV(GNU,EFDPTH,RECALF,MOL,FF,SF,VSF,XF,XS,XVS,XD,
     C     ZETAI,IZETA)
C
      IF(IPANEL.EQ.0) GO TO 180
C
150   CALL PANEL (FF,SF,VSF,KFILE)
C
      CALL SECOND (TIME)
      KPANEL=KPANEL+1
      IF (ISTOP.NE.1) GO TO 140
      END FILE KFILE
160   CONTINUE
      PRINT 955,TIME,TIMRDF,TIMCNV,TIMPNL
      IF (ICNT.LE.0) GO TO 170
      AVALF=SUMALF/FLOAT(ICNT)
      AVZETA=SUMZET/FLOAT(ICNT)
      PRINT 960,AVALF,AVZETA,ICNT,NCHNG
C
170   RETURN
C
C
C
900   FORMAT (1H0)
904   FORMAT(1X,7A10)
910   FORMAT(*0 OUTPUT FILE NO. =* I5)
915   FORMAT(*0 SECANT =* F15.5)
920   FORMAT (10(1PE10.3))
925   FORMAT(*0 PRESS(MB) =* F12.5/ *0 TEMP(K) =* F11.2)
930   FORMAT(33H0 COLUMN DENSITY(MOLECULES/CM**2)  ,
     1  /  /(5X,A6, * = * 1PE10.3) )
940   FORMAT(*0 DV(CM-1) = *F12.8/*0 V1(CM-1) = *F12.4/*0 V2(CM-1) = *
     1  F12.4)
942   FORMAT(17H0 BOUNDVS(CM-1) = ,F8.4)
945   FORMAT (* ----------------------------*F10.4,F14.7,3F10.6,I5)
950   FORMAT (* ++++++++++++++++++++++++++++*F10.4,E14.3,3F10.6,I5)
955   FORMAT (1H0,10X,*TIME*,11X,*READ*,4X,*CONVOLUTION*,10X,*PANEL*/
     1 4F15.3)
960   FORMAT (*0  AVERAGE WIDTH = *F10.5,*   AVERAGE ZETA = *F10.5,
     C *   NO  LINES = *I10, *    NO. CHANGES = *I10  )
970   FORMAT(1H0,5X,A10)
      END
```

```
              SUBROUTINE ROFILE (GNU,S,ALFA0,EPP,MOL)
              COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
              COMMON /BUF2/ VMIN,VMAX,NREC
              DIMENSION GNU(1), S(1), ALFA0(1), EPP(1), MOL(1)
              DATA LINFIL /3/, I1 /1/, I2 /2/
              IDATA=0
      $ 10      READ (LINFIL) VMIN,VMAX,NREC
      $         IF (EOF(LINFIL) .NE. 0 ) GO TO 80
      10      BUFFER IN (LINFIL,1) (VMIN,NREC)
              IF (UNIT(LINFIL) .EQ. 0 ) GO TO 80
              IF (NREC.GT.LIMIN) STOP " LIMIN "
              IF (VMAX.GE.VBOT) GO TO 30
      $         READ (LINFIL) VMIN
              BUFFER IN (LINFIL,1) ( GNU(I1),GNU(I2) )
              IF (UNIT(LINFIL) .EQ. 0 ) STOP
              GO TO 10
      $ 30      READ (LINFIL) (GNU(I),S(I),ALFA0(I),EPP(I),MOL(I),I=1,NREC)
      30      BUFFER IN (LINFIL,1) ( GNU(I1),MOL(LIMIN) )
              IF (UNIT(LINFIL) .EQ. 0 ) STOP
              ILO=1
              IHI=NREC
              IF (VMIN.GE.VBOT) GO TO 50
              DO 40 I=1,NREC
              ILO=I
              IF (GNU(I).GE.VBOT) GO TO 50
      40      CONTINUE
      50      IF (VMAX.LE.VTOP) RETURN
              DO 60 I=1,NREC
              IHI=I
              IF (GNU(I).GT.VTOP) GO TO 70
      60      CONTINUE
      70      IF(IHI.LT.NREC) IDATA=1
              RETURN
      80      PRINT 920
              IEOF=1
              RETURN
      C
      C
      920     FORMAT (* END OF FILE ON DISK*)
              END
```



```
               SUBROUTINE CONVFNV(GNU,EFDPTH,RECALF,MOL,FF,SF,VSF,XF,XS,XVS,XD,
              C     ZETAI,IZETA)
               COMMON/NFW/      NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
              C              DV,V1,V2,NLAYER
               COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
               COMMON /WXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
              1AX
               COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
               COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
               COMMON/VOICOM/AVRAT(201),CLD(201)
               DIMENSION GNU(1), EFDPTH(1), RECALF(1), MOL(1)
               DIMENSION FF(1), SF(1), VSF(1)
               DIMENSION XF(1), XS(1), XVS(1)
               DIMENSION XD(1)
               DIMENSION IZETA(1),ZETAI(1)
               CALL SECOND (TIME0)
               RATVX=DVVS/DXVS
               FVS=FLOAT(NWVS)/DXVS
               CONF=DVVS/DV-1.
               CONS=DVVS/DVS-1.
               IF (ILO.GT.IHI) GO TO 50
               DO 40 I=ILO,IHI
               DEPTHI=EFDPTH(I)
               IF(DEPTHI.LE.0.) GO TO 40
               IZ=IZETA(I)
               STRL=DEPTHI*CLD(IZ)
               STRD=DEPTHI*(1.-CLD(IZ))
```

```
      ZSLOPE=RECALF(I)*WATVN
      ZINT=(RNU(I)-VFT)/DVVS
      BFVS=FVS/ZSLOPE
      JMAX=(ZINT+BFVS)+2.
      IF (JMAX.LE.MAXVS) GO TO 20
      ILAST=I-1
      GO TO 50
 20   JMIN=(ZINT-BFVS)+2.
      JMINF=(ZINT*CONF)
      JMINS=(ZINT*CONS)
      ZVS=(FLOAT(JMIN-2)-ZINT)*ZSLOPE
      ZS=ZVS+(FLOAT(JMINS)-ZINT*CONS)*ZSLOPE
      ZF=ZVS+(FLOAT(JMINF)-ZINT*CONF)*ZSLOPE
      DO 30 JJ=JMIN,JMAX
      JS=JMINS+JJ
      JF=JMINF+JJ
      ZVS=ZVS+ZSLOPE
      ZS=ZS+ZSLOPE
      ZF=ZF+ZSLOPE
      IZVS=ABS(ZVS)+1.5
      IZS=ABS(ZS)+1.5
      IZF=ABS(ZF)+1.5
      VSF(JJ)=VSF(JJ)+STRL*XVS(IZVS)
      SF(JS)=SF(JS)+STRL*XS(IZS)
      FF(JF)=FF(JF)+STRL*XF(IZF)+STRD*XD(IZF)
 30   CONTINUE
 40   CONTINUE
      ILAST=IHI
C      IDATA=0 FOR MORE DATA REQUIRED ** IDATA=1 IF NO MORE DATA REQUIRED
      IPANEL=IDATA
      GO TO 60
 50   CONTINUE
      IPANEL=1
 60   ILO=ILAST+1
      CALL SECOND (TIME)
      TIMCNV=TIMCNV+TIME-TIME0
      RETURN
C
      END
```



```
      SUBROUTINE PANEL (FF,SF,VSF,KFILE)
      COMMON/NEW/     NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
     C             DV,V1,V2,NLAYER
      COMMON /XSUB/LIMIN,ILC,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
      COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
      COMMON /XPANEL/ V1P,V2P,DVP,NLIM,NSHIFT,NPTS
      COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
      DIMENSION FF(1), SF(1), VSF(1)
      DIMENSION SFSTOR(8)
      CALL SECOND (TIME0)
      X00=-7./128.
      X01=105./128.
      X02=35./128.
      X03=-5./128.
      X10=-1./16.
      X11=9./16.
      ISTOP=0
      NNHI=(V2-VFT)/DV+1.5
      IF (NHI.GE.NNHI) ISTOP=1
      IF (ISTOP.EQ.1) NHI=NNHI
      JMXPR=NLO+NPTS
      JJPR=NHI-NPTS
      LIMLO=(NLO-1)/4-3
      LIMHI=(NHI/4)+1
C     SLOW FUNCTION VALUES NEEDED FOR SUBSEQUENT PANELS ARE SAVED
      DO 10 J=1,8
 10   SFSTOR(J)=SF(LIMHI+J-5)
      DO 20 J=LIMLO,LIMHI,4
      JVS=(J-1)/4+1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      SF(J)=SF(J)+VSF(JVS)
      SF(J+1)=SF(J+1)+X00*VSF(JVS-1)+X01*VSF(JVS)+X02*VSF(JVS+1)+X03*VSF
     1(JVS+2)
      SF(J+2)=SF(J+2)+X10*(VSF(JVS-1)+VSF(JVS+2))+X11*(VSF(JVS)+VSF(JVS+
     11))
      SF(J+3)=SF(J+3)+X03*VSF(JVS-1)+X02*VSF(JVS)+X01*VSF(JVS+1)+X00*VSF
     1(JVS+2)
   20 CONTINUE
      DO 30 J=NLO,NHI,4
      JS=(J-1)/4+1
      FF(J)=FF(J)+SF(JS)
      FF(J+1)=FF(J+1)+X00*SF(JS-1)+X01*SF(JS)+X02*SF(JS+1)+X03*SF(JS+2)
      FF(J+2)=FF(J+2)+X10*(SF(JS-1)+SF(JS+2))+X11*(SF(JS)+SF(JS+1))
      FF(J+3)=FF(J+3)+X03*SF(JS-1)+X02*SF(JS)+X01*SF(JS+1)+X00*SF(JS+2)
   30 CONTINUE
      IF (NPTS.EQ.0) GO TO 60
      DO 40 J=NLO,JHXPR
      VP=VFT+FLOAT(J-1)*DV
   40 PRINT 900, J,FF(J),VP
      DO 50 J=JJPR,NHI
      VP=VFT+FLOAT(J-1)*DV
   50 PRINT 900, J,FF(J),VP
   60 CONTINUE
      DO 70 J=1,8
   70 SF(LIMHI+J-5)=SFSTOR(J)
      NLIM=NHI-NLO+1
      V1P=VFT+FLOAT(NLO-1)*DV
      V2P=VFT+FLOAT(NHI-1)*DV
C     V1P IS FIRST FREQ OF PANELV
C     V2P IS LAST FREQ OF PANELV
C     WRITE (KFILE) V1P,V2P,DVP,NLIM
C     WRITE (KFILE) (FF(J),J=NLO,NHI)
      BUFFER OUT (KFILE,1) (V1P,NLIM)
      IF (UNIT(KFILE) .EQ. 0 ) STOP
      BUFFER OUT (KFILE,1) (FF(NLO),FF(NHI))
      IF (UNIT(KFILE) .EQ. 0 ) STOP
      VFT=VFT+FLOAT(NLIMF-1)*DV
      IF (ISTOP.EQ.1) GO TO 140
      JF=1
      DO 80 J=NLIMF,MAXF
      FF(JF)=FF(J)
   80 JF=JF+1
      DO 90 J=JF,MAXF
   90 FF(J)=0.
      JS=1
      DO 100 J=NLIMS,MAXS
      SF(JS)=SF(J)
  100 JS=JS+1
      DO 110 J=JS,MAXS
  110 SF(J)=0.
      JVS=1
      DO 120 J=NLIMVS,MAXVS
      VSF(JVS)=VSF(J)
  120 JVS=JVS+1
      DO 130 J=JVS,MAXVS
  130 VSF(J)=0.
      NLO=NSHIFT+1
  140 CALL SECOND (TIME)
      TIMPNL=TIMPNL+TIME-TIME0
      RETURN
C
C
C
  900 FORMAT (I10,24X,E12.5,F12.5)
      END
```

```
      SUBROUTINE MOLEC(IND,NMOLID,NMOLEC,TEMP0,TEMP,PV,P,SCOR,ALFCOR,
     C ALFD1)
      DIMENSION NMOLID(1),SCOR(1),ALFCOR(1),ALFD1(1)
      DIMENSION W(7,4),ND(7,4),SMASS(7)
      COMMON /XMOLEC/ NM(7),NV(7),W1(7),N1(7),W2(7),N2(7),
     1  W3(7),N3(7),W4(7),N4(7),QV0(7),ROTFAC(7)
C
      DATA AVOG/6.0249E 23/, BOLTZ/1.3805E-16/, CLIGHT/2.997925E 10/
C
      DATA NM,SMASS,NV,W1,N1,W2,N2,W3,N3,W4,N4,QV0,ROTFAC  /
C
     1 6H   H2O ,6H   CO2 ,6H    O3 ,6H   N2O ,6H    CO ,6H   CH4 ,6H    O2 ,
     2      18. ,      44. ,      48. ,      44. ,      28. ,      16. ,      32. ,
     3        3 ,        3 ,        3 ,        3 ,        1 ,        4 ,        1 ,
     4  3651.7 , 1384.0 , 1103.2 , 1284.9 , 2143.3 , 2914.  , 1485.3 ,
     5        1 ,        1 ,        1 ,        1 ,        1 ,        1 ,        1 ,
     6  1595.0 ,  667.3 ,  700.9 ,  588.8 ,     0. , 1533.3 ,     0. ,
     7        1 ,        2 ,        1 ,        2 ,        0 ,        2 ,        0 ,
     8  3755.8 , 2349.3 , 1042.1 , 2223.8 ,     0. , 3018.9 ,     0. ,
     9        1 ,        1 ,        1 ,        1 ,        0 ,        3 ,        0 ,
     A      0. ,      0. ,      0. ,      0. ,      0. , 1297.9 ,      0. ,
     B        0 ,        0 ,        0 ,        0 ,        0 ,        3 ,        0 ,
     C  1.0004 , 1.0040 , 1.0456 , 1.1267 , 1.0000 , 1.0068 , 1.0007 ,
     D      1.5 ,     1.0 ,     1.5 ,     1.0 ,     1.0 ,     1.5 ,     1.0 /
C
C
C     MOLEC MAKES THE MOLECULAR IDENTIFICATIONS
C
C     SCOR IS THE FACTOR BY WHICH THE LINE INTENSITY IS CHANGED DUE TO
C        TEMPERATURE DEPENDENCE OF THE VIB AND ROT PARTITION SUMS
C
C
C     ALFCOR IS THE FACTOR BY WHICH THE COLLISION WIDTH MUST BE CHANGED
C        DUE TO THE DEPENDENCE ON PRESSURE AND TEMPERATURE
C        THE TEMPERATURE DEPENDENCE IS TAKEN AS (T0/T)**0.5
C
C
C     ALFD1 CONTAINS THE DOPPLER WIDTHS AT 1 CM-1
C
      IF (IND.EQ.2) GO TO 20
      NDIM=7
      NVDIM=4
      DO 10 M=1,NMOLEC
      W(M,1)=W1(M)
      W(M,2)=W2(M)
      W(M,3)=W3(M)
      W(M,4)=W4(M)
      ND(M,1)=N1(M)
      ND(M,2)=N2(M)
      ND(M,3)=N3(M)
      ND(M,4)=N4(M)
      FLN2=ALOG(2.)
   10 NMOLID(M)=NM(M)
      FAD=FLN2*2.*AVOG*BOLTZ/(CLIGHT*CLIGHT)
      RETURN
   20 CONTINUE
      PRATIO=P/P0
      TRATIO=TEMP0/TEMP
      XKT=0.6951*TEMP
      DO 30 M=1,NMOLEC
      SCOR(M)=QVRFAC(M,XKT,TRATIO,QV0,ROTFAC,W,ND,NDIM,NVDIM)
      ALFCOR(M)=PRATIO*(TRATIO**0.5)
      ALFD1(M)=SQRT(FAD*TEMP/SMASS(M))
   30 CONTINUE
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

FUNCTION OVRFAC   74/74   OPT=1   FTN 4.6+428   03/14/78

```
      FUNCTION OVRFAC(M,XKT,TRATIO,OVB,ROTFAC,N,ND,MDIM,NVDIM)
      DIMENSION OVB(1),ROTFAC(1)
      DIMENSION N(MDIM,NVDIM),ND(MDIM,NVDIM)
      OV=1.
      DO 10 I=1,NVDIM
      IF (ND(M,I).EQ.0) GO TO 20
      SV=1.-EXP(-N(M,I)/XKT)
      IF (ND(M,I).GT.1) SV=SV**ND(M,I)
   10 OV=OV/SV
   20 OVRFAC=(OVB(M)/OV)*((TRATIO)**ROTFAC(M))
      RETURN
C
      END
```

SUBROUTINE SHAPEL   74/74   OPT=1   FTN 4.6+428   03/14/78

```
      SUBROUTINE SHAPEL(XF,XS,XVS)
      COMMON /WXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
     1AX
      DIMENSION XF(1), XS(1), XVS(1)
      XLORNZ(X2)=1./(1.+X2)
      SFN(X2)=A1+B1*X2+C1*X2*X2
      VSFN(X2)=A2+B2*X2+C2*X2*X2
      A0(Z0)=(1.+3.*Z0**2+3.*Z0**4)/(1.+Z0**2)**3
      B0(Z0)=-(1.+3.*Z0**2)/(1.+Z0**2)**3
      C0(Z0)=1./(1.+Z0**2)**3
C     MATCH AT Z1 HALFWIDTHS
      Z1=4.
      A1=A0(Z1)
      B1=B0(Z1)
      C1=C0(Z1)
C     MATCH AT Z2 HALFWIDTHS
      Z2=16.
      A2=A0(Z2)
      B2=B0(Z2)
      C2=C0(Z2)
      RECPI=1./(2.*ASIN(1.))
      TOTAL=0.
      DO 10 I=1,NFMAX
   10 XF(I)=0.
      XF(1)=RECPI*(XLORNZ(0.)-SFN(0.))
      SUM=XF(1)
      DO 20 JJ=2,NF
      X=FLOAT(JJ-1)*DXF
      X2=X*X
      XF(JJ)=RECPI*(XLORNZ(X2)-SFN(X2))
      SUM=SUM+XF(JJ)*2.
   20 CONTINUE
      XF(NF)=0.
      SUM=SUM*DXF
      TOTAL=TOTAL+SUM
      DO 30 I=1,NSMAX
   30 XS(I)=0.
      XS(1)=RECPI*(SFN(0.)-VSFN(0.))
      SUM=XS(1)
      NS1=FLOAT(NWF)/DXS+1.001
      DO 40 JJ=2,NS1
      X=FLOAT(JJ-1)*DXS
      X2=X*X
      XS(JJ)=RECPI*(SFN(X2)-VSFN(X2))
      SUM=SUM+XS(JJ)*2.
   40 CONTINUE
      NS1P=NS1+1
      DO 50 JJ=NS1P,NS
      X=FLOAT(JJ-1)*DXS
      X2=X*X
      XS(JJ)=RECPI*(XLORNZ(X2)-VSFN(X2))
      SUM=SUM+XS(JJ)*2.
   50 CONTINUE
      XS(NS)=0.
```

```
      SUM=SUM*DXS
      TOTAL=TOTAL+SUM
      DO 60 I=1,NVSMAX
   60 XVS(I)=0.
      XVS(1)=RECPI*VSFN(0.)
      SUM=XVS(1)
      NVS1=FLOAT(NHS)/DXVS+1.001
      DO 70 JJ=2,NVS1
      X=FLOAT(JJ-1)*DXVS
      X2=X*X
      XVS(JJ)=RECPI*VSFN(X2)
      SUM=SUM+XVS(JJ)*2.
   70 CONTINUE
      NS1P=NVS1+1
      DO 80 JJ=NS1P,NVS
      X=FLOAT(JJ-1)*DXVS
      X2=X*X
      XVS(JJ)=RECPI*(XLORNZ(X2))
      SUM=SUM+XVS(JJ)*2.
   80 CONTINUE
      SUM=SUM*DXVS
      TOTAL=TOTAL+SUM
      RETURN
C
C
C
      END
```



```
      SUBROUTINE SHAPED(XD)
      COMMON /WXX/ NNF,DXF,NF,NHS,DXS,NS,NNVS,DXVS,NVS,NFMAX,NSMAX,NVSM
     1AX
      DIMENSION XD(1)
      XGAUSS(X2)=EXP(-1.*FLN2*X2)
      FLN2=ALOG(2.)
      RECPI=1./(2.*ASIN(1.))
      XDNORM=SQRT(FLN2*RECPI)
      TOTAL=0.
      DO 10 I=1,NFMAX
   10 XD(I)=0.
      XD(1)=XDNORM*XGAUSS(0.)
      SUM=XD(1)
      DO 20 JJ=2,NF
      X=FLOAT(JJ-1)*DXF
      X2=X*X
      XD(JJ)=XDNORM*XGAUSS(X2)
      SUM=SUM+XD(JJ)*2.
   20 CONTINUE
      XD(NF)=0.
      SUM=SUM*DXF
      TOTAL=TOTAL+SUM
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      SUBROUTINE VOICON
      COMMON /VOICON/ AVRAT(201),CLD(201)
C     LORENTZ AND DOPPLER FUNCTIONS DECOMPOSED OVER THE INTERVALS
C     0-4 , 0-16 , 0-64
C     WEIGHTING FUNCTIONS WAS   1. / VOIGT
      DATA (AVRAT(I),I=1,40) /
     C .1000000E+01,  .1002600E+01,  .1005400E+01,  .1008107E+01,
     C .1010949E+01,  .1013772E+01,  .1016629E+01,  .1019922E+01,
     C .1022451E+01,  .1025418E+01,  .1028422E+01,  .1031469E+01,
     C .1034547E+01,  .1037669E+01,  .1040831E+01,  .1044033E+01,
     C .1047281E+01,  .1050571E+01,  .1053904E+01,  .1057280E+01,
     C .1060798E+01,  .1064176E+01,  .1067693E+01,  .1071259E+01,
     C .1074679E+01,  .1078541E+01,  .1082290E+01,  .1086027E+01,
     C .1089858E+01,  .1093728E+01,  .1097601E+01,  .1101652E+01,
     C .1105700E+01,  .1109808E+01,  .1113970E+01,  .1110200E+01,
     C .1122499E+01,  .1126857E+01,  .1131281E+01,  .1135772E+01
     C /
      DATA (AVRAT(I),I=41,80) /
     C .1140332E+01,  .1144962E+01,  .1149665E+01,  .1154441E+01,
     C .1159292E+01,  .1164221E+01,  .1169220E+01,  .1174319E+01,
     C .1179405E+01,  .1184739E+01,  .1190080E+01,  .1195509E+01,
     C .1201028E+01,  .1206640E+01,  .1212347E+01,  .1218151E+01,
     C .1224054E+01,  .1230059E+01,  .1236169E+01,  .1242380E+01,
     C .1248712E+01,  .1255151E+01,  .1261705E+01,  .1268370E+01,
     C .1275172E+01,  .1282090E+01,  .1289136E+01,  .1296313E+01,
     C .1303625E+01,  .1311074E+01,  .1318666E+01,  .1326403E+01,
     C .1334290E+01,  .1342330E+01,  .1350520E+01,  .1358889E+01,
     C .1367416E+01,  .1376115E+01,  .1384991E+01,  .1394040E+01
     C /
      DATA (AVRAT(I),I=81,120) /
     C .1403291E+01,  .1412727E+01,  .1422360E+01,  .1432197E+01,
     C .1442244E+01,  .1452906E+01,  .1462991E+01,  .1473735E+01,
     C .1484654E+01,  .1495648E+01,  .1507292E+01,  .1518999E+01,
     C .1530965E+01,  .1543210E+01,  .1555740E+01,  .1568963E+01,
     C .1581669E+01,  .1595127E+01,  .1608890E+01,  .1622906E+01,
     C .1637427E+01,  .1619509E+01,  .1642006E+01,  .1584930E+01,
     C .1562C59E+01,  .1551887E+01,  .1535943E+01,  .1520369E+01,
     C .1505144E+01,  .1490272E+01,  .1475730E+01,  .1461539E+01,
     C .1447664E+01,  .1434105E+01,  .1420857E+01,  .1407913E+01,
     C .1395268E+01,  .1382908E+01,  .1370836E+01,  .1359042E+01
     C /
      DATA (AVRAT(I),I=121,160) /
     C .1347921E+01,  .1336288E+01,  .1325277E+01,  .1314543E+01,
     C .1304062E+01,  .1293828E+01,  .1283837E+01,  .1274084E+01,
     C .1264566E+01,  .1255277E+01,  .1246215E+01,  .1237374E+01,
     C .1228751E+01,  .1220343E+01,  .1212145E+01,  .1204155E+01,
     C .1196360E+01,  .1188782E+01,  .1181393E+01,  .1174197E+01,
     C .1167193E+01,  .1160376E+01,  .1153745E+01,  .1147295E+01,
     C .1141024E+01,  .1134929E+01,  .1129008E+01,  .1123238E+01,
     C .1117675E+01,  .1112259E+01,  .1107004E+01,  .1101910E+01,
     C .1096974E+01,  .1092192E+01,  .1087564E+01,  .1083085E+01,
     C .1078755E+01,  .1074570E+01,  .1070527E+01,  .1066626E+01
     C /
      DATA (AVRAT(I),I=161,201) /
     C .1062862E+01,  .1059236E+01,  .1055740E+01,  .1052377E+01,
     C .1049143E+01,  .1046035E+01,  .1043052E+01,  .1040190E+01,
     C .1037449E+01,  .1034824E+01,  .1032315E+01,  .1029919E+01,
     C .1027633E+01,  .1025456E+01,  .1023396E+01,  .1021419E+01,
     C .1019555E+01,  .1017790E+01,  .1016122E+01,  .1014550E+01,
     C .1013072E+01,  .1011684E+01,  .1010386E+01,  .1009179E+01,
     C .1008049E+01,  .1007006E+01,  .1006044E+01,  .1005160E+01,
     C .1004354E+01,  .1003623E+01,  .1002965E+01,  .1002378E+01,
     C .1001860E+01,  .1001410E+01,  .1001026E+01,  .1000709E+01,
     C .1000447E+01,  .1000249E+01,  .1000110E+01,  .1000027E+01,
     C 1. /
      DATA (CLD(I),I=1,40) /
     C .1460247E-20,  .6069322E-02,  .1239734E-01,  .1882509E-01,
     C .2531673E-01,  .3185520E-01,  .3843082E-01,  .4503691E-01,
     C .5166904E-01,  .5832309E-01,  .6499891E-01,  .7169210E-01,
     C .7840185E-01,  .8512682E-01,  .9186509E-01,  .9861810E-01,
     C .1053826E+00,  .1121580E+00,  .1189458E+00,  .1257433E+00,
     C .1325506E+00,  .1393673E+00,  .1461929E+00,  .1530277E+00,
     C .1598692E+00,  .1667192E+00,  .1735766E+00,  .1804411E+00,
     C .1873124E+00,  .1941901E+00,  .2010739E+00,  .2079636E+00,
```

```
     C  .2140588E+00,  .2217593E+00,  .2286867E+00,  .2355745E+00,
     C  .2424897E+00,  .2494078E+00,  .2563301E+00,  .2632559E+00
     C  /
      DATA (CLD(I),I=41,80) /
     C  .2701850E+00,  .2771169E+00,  .2840515E+00,  .2909883E+00,
     C  .2979271E+00,  .3048678E+00,  .3118094E+00,  .3187523E+00,
     C  .3256958E+00,  .3326396E+00,  .3395835E+00,  .3465269E+00,
     C  .3534697E+00,  .3604113E+00,  .3673519E+00,  .3742898E+00,
     C  .3812258E+00,  .3881592E+00,  .3950895E+00,  .4020163E+00,
     C  .4089391E+00,  .4158578E+00,  .4227712E+00,  .4296798E+00,
     C  .4365821E+00,  .4434784E+00,  .4503688E+00,  .4572503E+00,
     C  .4641248E+00,  .4709910E+00,  .4778403E+00,  .4846962E+00,
     C  .4915342E+00,  .4983646E+00,  .5051779E+00,  .5119822E+00,
     C  .5187743E+00,  .5255513E+00,  .5323187E+00,  .5390698E+00
     C  /
      DATA (CLD(I),I=81,120) /
     C  .5458059E+00,  .5525262E+00,  .5592302E+00,  .5659170E+00,
     C  .5725860E+00,  .5792364E+00,  .5858675E+00,  .5924784E+00,
     C  .5990684E+00,  .6056366E+00,  .6121823E+00,  .6187046E+00,
     C  .6252026E+00,  .6316756E+00,  .6381225E+00,  .6445428E+00,
     C  .6509349E+00,  .6572984E+00,  .6636323E+00,  .6699355E+00,
     C  .6762872E+00,  .6824463E+00,  .6886510E+00,  .6948228E+00,
     C  .7009582E+00,  .7070569E+00,  .7131188E+00,  .7191403E+00,
     C  .7251229E+00,  .7310646E+00,  .7369645E+00,  .7428212E+00,
     C  .7486337E+00,  .7544009E+00,  .7601218E+00,  .7657951E+00,
     C  .7714198E+00,  .7769948E+00,  .7825188E+00,  .7879908E+00
     C  /
      DATA(CLD(I),I=121,160) /
     C  .7934097E+00,  .7987743E+00,  .8040834E+00,  .8093360E+00,
     C  .8145309E+00,  .8196669E+00,  .8247431E+00,  .8297583E+00,
     C  .8347114E+00,  .8396014E+00,  .8444271E+00,  .8491876E+00,
     C  .8538818E+00,  .8585088E+00,  .8630675E+00,  .8675570E+00,
     C  .8719763E+00,  .8763246E+00,  .8806011E+00,  .8848047E+00,
     C  .8889349E+00,  .8929907E+00,  .8969715E+00,  .9008766E+00,
     C  .9047053E+00,  .9084571E+00,  .9121313E+00,  .9157274E+00,
     C  .9192450E+00,  .9226837E+00,  .9260431E+00,  .9293229E+00,
     C  .9325228E+00,  .9356426E+00,  .9386822E+00,  .9416414E+00,
     C  .9445203E+00,  .9473188E+00,  .9500370E+00,  .9526751E+00
     C  /
      DATA (CLD(I),I=161,201) /
     C  .9552332E+00,  .9577116E+00,  .9601105E+00,  .9624304E+00,
     C  .9646716E+00,  .9668346E+00,  .9689199E+00,  .9709282E+00,
     C  .9728599E+00,  .9747158E+00,  .9764967E+00,  .9782032E+00,
     C  .9798361E+00,  .9813965E+00,  .9828850E+00,  .9843028E+00,
     C  .9856506E+00,  .9869297E+00,  .9881409E+00,  .9892853E+00,
     C  .9903642E+00,  .9913784E+00,  .9923294E+00,  .9932181E+00,
     C  .9940458E+00,  .9948137E+00,  .9955230E+00,  .9961749E+00,
     C  .9967707E+00,  .9973117E+00,  .9977990E+00,  .9982340E+00,
     C  .9986179E+00,  .9989519E+00,  .9992374E+00,  .9994755E+00,
     C  .9996676E+00,  .9998148E+00,  .9999185E+00,  .9999798E+00,
     C  1.  /
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE FROM COPY FURNISHED TO DDC


```
      SUBROUTINE ABSMRG (ITYPE,ISMALL,IOLDPL,LFILE,MFILE)
      COMMON    FF(2610),DUMF(2610)
      COMMON /MAIN/ KFILE,KPANEL,P0,TEMP0
      COMMON /XABS/ NHOL,XID(7),SECANT,PAVE,TAVE,HHOLID(7),HK(7),
     C           DV,V1,V2,NLAYER
      DIMENSION SAVEDD(5), A1(10), A2(10), A3(10), A4(10)
      COMMON /OPANL/ OV1P,OV2P,ODVP,NLIMO
      COMMON /NPANL/ V1P,V2P,DVP,NLIM
      ISTOP=0
      CALL SECOND (TIME)
      PRINT 220, TIME
      INNL=1
      ISAVE=0
      SAVEDD(4)=0.
C
C ISMALL=1, IF DV(K-1) IS LESS THAN DV(K)
C
      IF (ISMALL.EQ.1) GO TO 10
      IOLD=LFILE
      KNEW=KFILE
      KNL=KPANEL
      GO TO 20
   10 KNEW=LFILE
      IOLD=KFILE
      KNL=IOLDPL
   20 REWIND 11
      PRINT*,"MFILE,LFILE,KFILE,KPANEL ",MFILE,LFILE,KFILE,KPANEL
      REWIND 12
      REWIND KFILE
      SAVED1=0.0
$     READ(KFILE)(XID(I),I=1,7),SECANT,PAVE,TAVE,(HHOLID(I),I=1,7),
$    1(HK(I),I=1,7),DV,V1,V2,NLAYER
$     WRITE(MFILE)(XID(I),I=1,7),SECANT,PAVE,TAVE,(HHOLID(I),I=1,7),
$    1(HK(I),I=1,7),DV,V1,V2,NLAYER
$     READ (LFILE) XID(1),XID(2)
      BUFFER IN(KFILE,1)(XID(1),XID(2))
      IF(UNIT(KFILE) .EQ. 0) STOP" 2"
  260 BUFFER OUT(MFILE,1)(XID(1),NLAYER)
      IF(UNIT(MFILE) .EQ. 0) STOP" 1"
  420 BUFFER IN(LFILE,1)(XID(1),NLAYER)
      IF(UNIT(LFILE) .EQ. 0) STOP" 3"
  200 DO 30 K=1,5
   30 SAVEDD(K)=0.
      ATYPE=ITYPE
      AP=1.0/(ATYPE+1.0)
      IF(ITYPE .NE. 0) GO TO 430
C
C 1/1 RATIO ONLY
C
      DO 435 KOD=1,KNL
      BUFFER IN(IOLD,1)(OV1P,NLIMO)
      IF(UNIT(IOLD) .EQ. 0) STOP"12"
      BUFFER IN(IOLD,1)(DUMF(1),DUMF(NLIMO))
      IF(UNIT(IOLD) .EQ. 0) STOP"13"
      BUFFER IN(KNEW,1)(V1P,NLIM)
      IF(UNIT(KNEW) .EQ. 0) STOP"14"
      BUFFER IN(KNEW,1)(FF(1),FF(NLIM))
      IF(UNIT(KNEW) .EQ. 0) STOP"15"
      DO 440 KOD=1,NLIM
  440 FF(KOD)=FF(KOD)+DUMF(KOD)
      BUFFER OUT(MFILE,1)(V1P,NLIM)
      IF(UNIT(MFILE) .EQ. 0) STOP"16"
      BUFFER OUT(MFILE,1)(FF(1),FF(NLIM))
      IF(UNIT(MFILE) .EQ. 0) STOP"17"
  435 CONTINUE
      GO TO 210
C
C ALL RATIOS EXCEPT 1/1
C
  430 LL=ITYPE+1
      DO 40 JPG=1,ITYPE
      APG=JPG
      P=1.0-(AP*APG)
```

```
      C
      C***** THE FOLLOWING ARE THE CONSTANTS FOR THE LAGRANGE 4 POINT
      C INTERPOLATION.  ********
      C
             A1(JPG)=-P*(P-1.0)*(P-2.0)/6.0
             A2(JPG)=(P**2-1.0)*(P-2.0)*0.5
             A3(JPG)=-P*(P+1.0)*(P-2.0)*0.5
             A4(JPG)=P*(P**2-1.0)/6.0
      40     CONTINUE
      C
      C ********  BEGINNING OF LOOP THAT DOES ADDITION  *********
      C
      $      READ(IOLD) OV1P,OV2P,ODVPP,NLIMO
             BUFFER IN(IOLD,1)(OV1P,NLIMO)
             IF(UNIT(IOLD)  EQ. 0) STOP" 4"
      $      READ(IOLD)(DUMF(J),J=1,NLIMO)
      310    BUFFER IN(IOLD,1)(DUMF(1),DUMF(NLIMO))
             IF(UNIT(IOLD) .EQ. 0) STOP" 5"
      320    NVS=1
      $      READ(KNEW) V1P,V2P,DVP,NLIM
             BUFFER IN(KNEW,1)(V1P,NLIM)
             IF(UNIT(KNEW) .EQ. 0) STOP" 6"
      $      READ(KNEW)(FF(J),J=1,NLIM)
      330    BUFFER IN(KNEW,1)(FF(1),FF(NLIM))
             IF(UNIT(KNEW) .EQ. 0) STOP" 7"
      340    FF(1)=FF(1)+DUMF(1)
             II=2
      60     JJ=1
             DO 170 JPG=1,LL
             IF (JPG.EQ.LL) GO TO 70
             IF (NVS.EQ.1) GO TO 80
             GO TO 90
      70     FF(II)=FF(II)+DUMF(NVS)
             GO TO 110
      80     IF(SAVEDD(4) .EQ. 0.0) GO TO 81
             DUMFY=SAVEDD(4)
             GO TO 100
      81     DUMFY=DUMF(1)
             GO TO 100
      90     DUMFY=DUMF(NVS-1)
      100    FF(II)=FF(II)+A1(JJ)*DUMFY+A2(JJ)*DUMF(NVS)+A3(JJ)*DUMF(NVS+1)+A4(
            1JJ)*DUMF(NVS+2)
      110    NVS=NVS+1
             IF (NVS.LE.NLIMO-2) GO TO 140
             SAVEDD(1)=DUMF(NVS-1)
             SAVEDD(2)=DUMF(NVS)
             SAVEDD(3)=DUMF(NVS+1)
             SAVEDD(4)=DUMF(NVS-2)
      $      READ(IOLD) OV1P,OV2P,ODVPP,NLIMO
      $      IF(EOF(IOLD)) 180,120
             BUFFER IN(IOLD,1)(OV1P,NLIMO)
             IF(UNIT(IOLD)) 120,180,120
      120    NLIMO=NLIMO+3
      $      READ(IOLD)(DUMF(J),J=4,NLIMO)
      $      IF(EOF(IOLD)) 180,130
             BUFFER IN(IOLD,1)(DUMF(4),DUMF(NLIMO))
             IF(UNIT(IOLD)) 130,180,130
      130    DUMF(1)=SAVEDD(1)
             DUMF(2)=SAVEDD(2)
             DUMF(3)=SAVEDD(3)
             NVS=2
      140    II=II+1
             JJ=JJ+1
             IF (II.GT.NLIM) GO TO 150
             GO TO 170
      180    II=II+1
             AVRG=(SAVEDD(3)+SAVEDD(2))*0.5
      200    FF(II)=FF(II)+AVRG
             II=II+1
             IF(II .LE. NLIM) GO TO 200
```

```
      C
      C WRITE OUTPUT FILE
      C
      $150  WRITE(MFILE) V1P,V2P,DVP,NLIM
      $     WRITE(MFILE)(FF(J),J=1,NLIM)
      150   BUFFER OUT(MFILE,1)(V1P,NLIM)
            IF(UNIT(MFILE) .EQ. 0) STOP" 8"
      370   BUFFER OUT(MFILE,1)(FF(1),FF(NLIM))
            IF(UNIT(MFILE) .EQ. 0) STOP" 9"
        380 IKNL=IKNL+1
            IF (IKNL.LE.KNL) GO TO 160
            GO TO 210
      $     READ(KNEW) V1P,V2P,DVP,NLIM
      160   BUFFER IN(KNEW,1)(V1P,NLIM)
            IF(UNIT(KNEW) .EQ. 0) STOP" 10"
      $     READ(KNEW)(FF(J),J=1,NLIM)
      390   BUFFER IN(KNEW,1)(FF(1),FF(NLIM))
            IF(UNIT(KNEW) .EQ. 0) STOP" 11"
      360   II=1
      170   CONTINUE
            NVS=NVS-1
            GO TO 60
      210   REWIND KFILE
      C
            CALL SECOND (TIME1)
            TIM=TIME1-TIME
            PRINT 240, TIME1,TIM
            RETURN
      C
      C
      C
      220   FORMAT (*0 THE TIME AT THE START OF ABSHRG IS *,F12.3)
      240   FORMAT (*0 THE TIME AT THE END OF ABSHRG IS*,F12.3/F12.3,* SECS WE
           1RE REQUIRED FOR THIS ADDITION*)
            END
```



```
            SUBROUTINE EMINIT(ILOOK,TBOUND,MFILE)
            COMMON EMISS(2410),BODIES(101),FF(2410),EDGE(101)
            COMMON/MAIN/ KFILE,KPANEL,PO,TEMPO
            COMMON/NEW/     NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),NK(7),
           C                DV,V1,V2,NLAYER
            COMMON /OPANL/ OV1P,OV2P,ODVP,NLIMO
      C
      C ****************************************************************
      C ******  THIS SUBROUTINE COMPUTES THE EMISSION FOR THE FIRST LAYER  ***
      C ****************************************************************
      C
      C TBOUND IS A TEMPERATURE USED IN COMPUTING A BLACK BODY WHICH
      C WILL BE USED AS A BOUNDRY.
      C
            EXPF(Y)=EXP(AMAX1(-174.0,Y))
            BBODY(X)=(1.1910E-12*X**3)/(EXPF(1.43789*X/TAVE)-1.0)
            BOUND(X)=(1.1910E-12*X**3)/(EXPF(1.43789*X/TBOUND)-1.0)
            PRINT *, " EMINIT, MFILE, KFILE  ", MFILE,KFILE
            CALL SECOND(TIME)
            SS=0.0
            REWIND MFILE
      $     READ(KFILE)(XID(I),I=1,7),SECANT,PAVE,TAVE,(HMOLID(I),I=1,7),
      $    1(NK(I),I=1,7),DV,V1,V2,NLAYER
      $     WRITE(MFILE)(XID(I),I=1,7),SECANT,PAVE,TAVE,(HMOLID(I),I=1,7),
      $    1(NK(I),I=1,7),DV,V1,V2,NLAYER
            BUFFER IN(KFILE,1)(XID(1),NLAYER)
            IF(UNIT(KFILE) .EQ. 0) STOP
            BUFFER OUT(MFILE,1)(XID(1),NLAYER)
            IF(UNIT(MFILE) .EQ. 0.) STOP
            DO 1 II=1,KPANEL
      $     READ(KFILE) OV1P,OV2P,ODVP,NLIMO
```

```
          BUFFER IN(KFILE,1)(OV1P,NLIMO)
          IF(UNIT(KFILE) .EQ. 0.) STOP
          BEGIN=OV1P
          OV=ODVP
          DOC=BEGIN
          I=1
   270    BODIES(I)=BBODY(BEGIN)
          IF(TBOUND .EQ. 0.) GO TO 40
          EDGE(I)=BOUND(BEGIN)
    40    I=I+1
          IF(I .GT. 101) STOP
          BEGIN=BEGIN+1.0
          IF(BEGIN .LE. OV2P+1.0) GO TO 270
          IF(I .EQ. 2) BODIES(2)=BODIES(1)
          COMON=1.44*OV1P/TAVE
          IF(COMON .LE. 4.605) GO TO 280
          DENOM=1.
          GO TO 290
   280    DENOM=1.0-EXPF(-COMON)
   290    DVBBF=OV1P*1.0E-4/ABS(3.0-(COMON/DENOM))
          BB=BODIES(1)
          BE=EDGE(1)
          IF(TBOUND .EQ. 0.) BE=0.
          TESTER=OV1P
          CHANGE=OV1P+DVBBF
          LTEST=1
     5    READ(KFILE)(FF(J),J=1,NLIMO)
          BUFFER IN(KFILE,1)(FF(1),FF(NLIMO))
          IF(UNIT(KFILE) .EQ. 0.) STOP
    20    DO 2 J=1,NLIMO
          FF(J)=EXPF(-FF(J))
          IF(ILOOK .EQ. 0) GO TO 30
     C
     C EMISSION LOOKING DOWN
     C
          EMISS(J)=(1.0-FF(J))*BB+FF(J)*BE
     C
     C
          GO TO 35
     C
     C EMISSION LOOKING UP
     C
    30    EMISS(J)=(1.0-FF(J))*BB
       35 TESTER=TESTER+DV
          IF(TESTER .LT. CHANGE) GO TO 2
          COMCN=CHANGE-BOD
          BB=BODIES(LTEST)-(BODIES(LTEST)-BODIES(LTEST+1))*COMCN
          IF(TBOUND .NE. 0) BE= EDGE(LTEST)-( EDGE(LTEST)- EDGE(LTEST+1)
         1)*COMON
          CHANGE=CHANGE+DVBBF
          IF(CHANGE .GE. BOD+1.0) GO TO 5
          GO TO 2
        5 LTEST=LTEST+1
          BOD=BOD+1.0
     2    CONTINUE
     $    WRITE(MFILE) OV1P,OV2P,ODVP,NLIMO
     $    WRITE(MFILE)(EMISS(J),J=1,NLIMO)
     $    WRITE(MFILE)(FF(J),J=1,NLIMO)
          BUFFER OUT(MFILE,1)(OV1P,NLIMO)
          IF(UNIT(MFILE) .EQ. 0.) STOP
          BUFFER OUT(MFILE,1)(EMISS(1),EMISS(NLIMO))
          IF(UNIT(MFILE) .EQ. 0.) STOP
          BUFFER OUT(MFILE,1)(FF(1),FF(NLIMO))
          IF(UNIT(MFILE) .EQ. 0.) STOP
     1    CONTINUE
          CALL SECOND(TIME1)
          TIME=TIME1-TIME
          PRINT *,TIME," SECS WERE NEEDED FOR EMINIT "
          RETURN
          END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      SUBROUTINE EMUP(NPTS,ITYPE,LFILE,MFILE)
      COMMON    OLDEM(2410),OLDTR(2410),NEWEM(2410)
      COMMON/MAIN/ KFILE,KPANEL,P0,TEMP0
      COMMON/XEMISS/     MMOL,XID(7),SECANT,PAVE,TAVE,MMOLID(7),WK(7),
     C               DV,V1,V2,NLAYER
      COMMON /OPANL/ OV1P,OV2P,ODVP,NLIMO
      DIMENSION EMSAVE(4),A1(10),A2(10),A3(10),A4(10),TRSAVE(4)
      DIMENSION TR(2410),NEWTR(1),BODIES(101)
      COMMON /NPANL/ V1P,V2P,DVP,NLIM
      EQUIVALENCE (TR(1),NEWTR(1))
      REAL NEWEM,NEWTR
C
C ****************************************************************
C ****** THIS SUBROUTINE DOES THE LAYER ADDITION FOR THE EMISSION  *****
C        LOOKING FROM GROUND TO SPACE.
C ****************************************************************
C
      EXPF(Y)=EXP(AMAX1(-174.0,Y))
      BBODY(X)=(1.1910E-12*X**3)/(EXPF(1.43789*X/TAVE)-1.0)
      EXPMIN=EXP(-174.0)
      CALL SECOND(TIME)
      PRINT *, " EMUP, MFILE, LFILE, KFILE  ", MFILE,LFILE,KFILE
      PRINT 220,TIME
      IKNL=1
      PRINT *," THIS OUTPUT IS WRITTEN ON FILE  ",MFILE
C
      KNL=KPANEL
20    REWIND MFILE
      REWIND LFILE
      REWIND KFILE
      BUFFER IN (LFILE,1) (XID(1),XID(2))
      IF (UNIT(LFILE) .EQ. 0 ) STOP
      BUFFER IN(KFILE,1)(XID(1),NLAYER)
      IF(UNIT(KFILE) .EQ. 0) STOP
      BUFFER OUT(MFILE,1)(XID(1),NLAYER)
      IF(UNIT(MFILE) .EQ. 0) STOP
      DO 30 K=1,4
      EMSAVE(K)=0.
30    TRSAVE(K)=0.
      ATYPE=ITYPE
      IF(ITYPE .NE. 0) GO TO 430
C
C 1/1 RATIO ONLY
C
      JUG=3
      JOG=2
      GO TO 360
  53? BUFFER IN(LFILE,1)(OV1P,NLIMO)
      IF(UNIT(LFILE) .EQ. 0) STOP"2"
      BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP"3"
      BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP"4"
      DO 530 I=1,NLIM
      NEWEM(I)=OLDEM(I)+(1.0-TR(I))*BB*OLDTR(I)
      NEWTR(I)=TR(I)*OLDTR(I)
      TESTER=TESTER+DVP
      IF(TESTER .LT. CHANGE) GO TO 530
      CONON=CHANGE-BOD
      BB=BODIES(LTEST)-(BODIES(LTEST)-BODIES(LTEST+1))*CONON
      CHANGE=CHANGE+DVARF
      IF(CHANGE .GE. BOD+1.0) GO TO 560
      GO TO 530
  560 LTEST=LTEST+1
      BOD=BOD+1.0
  530 CONTINUE
      GO TO 150
  590 IKNL=IKNL+1
      IF(IKNL .LE. KNL) GO TO 360
      GO TO 210
C
C ALL RATIOS EXCEPT 1/1
C
430   LL=ITYPE+1
```

```
      AP=1.0/(ATYPE+1.0)
      JOG=1
      DO 40 JPG=1,ITYPE
      APG=JPG
      P=1.0-(AP*APG)
      PRINT *," P= ",P
C
C ***** THE FOLLOW ARE THE CONSTANTS FOR THE LAGRANGE 4 POINT
C ***** INTERPOLATION
C
      A1(JPG)=-P*(P-1.0)*(P-2.0)/6.0
      A2(JPG)=(P**2-1.0)*(P-2.0)*0.5
      A3(JPG)=-P*(P+1.0)*(P-2.0)*0.5
      A4(JPG)=P*(P**2-1.0)/6.0
40    CONTINUE
C
C ***** BEGINNING OF LOOP THAT DOES ADDITION  *****
C
      JUG=1
$     READ(LFILE) OV1P,OV2P,ODVP,NLIMO
      BUFFER IN(LFILE,1)(OV1P,NLIMO)
      IF(UNIT(LFILE) .EQ. 0) STOP
C
C NEWEM IS THE REGION FOR THE EMISSION SUMMATION
C
$     READ(LFILE)(OLDEM(J),J=1,NLIMO)
      BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP
C
C NEWTR IS THE REGION FOR THE TRANSMISSION SUMMATION
C
      BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP
      NVS=1
      GO TO 360
340   NEWEM(1)=OLDEM(1)+(1.0-TR(1))*BB*OLDTR(1)
      NEWTR(1)=TR(1)*OLDTR(1)
      TESTER=TESTER+DVP
      II=2
60    JJ=1
C
C
      DO 190 JPG=1,LL
      IF(JPG .EQ. LL) GO TO 70
      IF(NVS .EQ. 1) GO TO 80
      GO TO 90
70    NEWEM(II)=OLDEM(NVS)+(1.0-TR(II))*BB*OLDTR(NVS)
      NEWTR(II)=TR(II)*OLDTR(NVS)
      TESTER=TESTER+DVP
      GO TO 110
80    IF(EMSAVE(4) .EQ. 0.0) GO TO 81
      EMSTOR=EMSAVE(4)
      TRSTOR=TRSAVE(4)
      GO TO 100
81    EMSTOR=OLDEM(1)
      TRSTOR=OLDTR(1)
      GO TO 100
90    EMSTOR=OLDEM(NVS-1)
      TRSTOR=OLDTR(NVS-1)
C
C INTERPOLATE FOR THE OLD EMISSION
C
100   OLDEMI=A1(JJ)*EMSTOR+A2(JJ)*OLDEM(NVS)+A3(JJ)*OLDEM(NVS+1)+
     1A4(JJ)*OLDEM(NVS+2)
C
C INTERPOLATE FOR THE OLD TRANSMISSION
C
      OLDTRI=A1(JJ)*TRSTOR+A2(JJ)*OLDTR(NVS)+A3(JJ)*OLDTR(NVS+1)
     1+A4(JJ)*OLDTR(NVS+2)
      NEWEM(II)=OLDEMI+(1.0-TR(II))*BB*OLDTRI
      NEWTR(II)=TR(II)*OLDTRI
      TESTER=TESTER+DVP
110   NVS=NVS+1
      IF(NVS .LE. NLIMO-2) GO TO 140
      EMSAVE(1)=OLDEM(NVS-1)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
              EMSAVE(2)=OLDEM(NVS)
              EMSAVE(3)=OLDEM(NVS+1)
              EMSAVE(4)=OLDEM(NVS-2)
              TRSAVE(1)=OLDTR(NVS-1)
              TRSAVE(2)=OLDTR(NVS)
              TRSAVE(3)=OLDTR(NVS+1)
              TRSAVE(4)=OLDTR(NVS-2)
      $       READ(LFILE) OV1P,OV2P,ODVP,NLIMO
      $       IF(EOF(LFILE)) 180,120
              BUFFER IN(LFILE,1)(OV1P,NLIMO)
              IF(UNIT(LFILE)) 120,180,120
      120     NLIMO=NLIMO+3
      $       READ(LFILE)(OLDEM(J),J=4,NLIMO)
              BUFFER IN(LFILE,1)(OLDEM(4),OLDEM(NLIMO))
              IF(UNIT(LFILE)) 130,180,130
      $130    READ(LFILE)(OLDTR(J),J=4,NLIMO)
      130     BUFFER IN(LFILE,1)(OLDTR(4),OLDTR(NLIMO))
              IF(UNIT(LFILE) .EQ. 0) STOP
              OLDEM(1)=EMSAVE(1)
              OLDEM(2)=EMSAVE(2)
              OLDEM(3)=EMSAVE(3)
              OLDTR(1)=TRSAVE(1)
              OLDTR(2)=TRSAVE(2)
              OLDTR(3)=TRSAVE(3)
              NVS=2
      140     II=II+1
              JJ=JJ+1
              IF(II .GT. NLIM) GO TO 150
              GO TO 170
      C
      C AN EOF HAS OCCURED ON THE FILE FOR THE PREVIOUS LAYER
      C
      180     II=II+1
              EM1=(EMSAVE(3)+EMSAVE(2))*0.5
              TR1=(TRSAVE(3)+TRSAVE(2))*0.5
      200     NEWEM(II)=EM1+(1.0-TR(II))*BB*TR1
              NEWTR(II)=TR(II)*TR1
              II=II+1
              IF(II .LE. NLIM) GO TO 200
      $150    WRITE(MFILE)V1P,V2P,DV,NLIM
      $       WRITE(MFILE)(NEWEM(J),J=1,NLIM)
      $       WRITE(MFILE)(NEWTR(J),J=1,NLIM)
      C
      C WRITE OUTPUT FILE
      C
        150 BUFFER OUT(MFILE,1)(V1P,NLIM)
              IF(UNIT(MFILE) .EQ. 0) STOP
              BUFFER OUT(MFILE,1)(NEWEM(1),NEWEM(NLIM))
              IF(UNIT(MFILE) .EQ. 0) STOP
              BUFFER OUT(MFILE,1)(NEWTR(1),NEWTR(NLIM))
              IF(UNIT(MFILE) .EQ. 0) STOP
              IF ( NPTS .LE. 0 ) GO TO 333
              DO 320 J=1,NPTS
              VP=V1P+FLOAT(J-1)*DVP
      320     PRINT 900, J,VP,NEWEM(J),NEWTR(J)
              JEND=NLIM-NPTS+1
              DO 330 J=JEND,NLIM
              VP=V1P+FLOAT(J-1)*DVP
      330     PRINT 900, J,VP,NEWEM(J),NEWTR(J)
      900     FORMAT(I10,F12.6,3E15.8)
      333     CONTINUE
              GO TO (545,550),JUG
        545 IKNL=IKNL+1
              IF(IKNL .LE. KNL) GO TO 160
              GO TO 210
      160     JUG=2
      360     BUFFER IN(KFILE,1)(V1P,NLIM)
              IF(UNIT(KFILE) .EQ. 0) STOP
              BUFFER IN(KFILE,1)(TR(1),TR(NLIM))
              IF(UNIT(KFILE) .EQ. 0) STOP
      C
      C COMPUTE THE BLACK BODY EVERY 1 WAVE # BEGINNING AT V1P
      C
```

```
            BEGIN=V1P
            BOD=BEGIN
            I=1
      270   BODIES(I)=BRODY(BEGIN)
            I=I+1
            IF(I .GT. I81) STOP
            BEGIN=BEGIN+1.0
            IF(BEGIN .LE. V2P+1.0) GO TO 270
            IF(I .EQ. 2)  BODIES(2)=BODIES(1)
            COMON=1.44*V1P/TAVE
            IF(COMON .LE. 4.605) GO TO 280
            DENOM=1.
            GO TO 290
      280   DENOM=1.0-EXPF(-COMON)
      290   DVBBF=V1P*1.0E-4/ABS(3.0-(COMON/DENOM))
      CCC     PRINT *,(BODIES(LUD),LUD=1,101)
            BB=BODIES(1)
            TESTER=V1P
            CHANGE=V1P+DVBBF
            LTEST=1
            DO 260 J=1,NLIM
            TRJ=TR(J)
            TR(J)=1.
            IF (TRJ .LE. 0. ) GO TO 260
            IF (TRJ .LT. 174.0 ) GO TO 258
            TR(J)=EXPMIN
            GO TO 260
      258   TR(J)=EXP(-TRJ)
      260   CONTINUE
            GO TO (340,350,535),JUG
      350   II=1
        170 IF(TESTER .LT. CHANGE) GO TO 190
            COMON=CHANGE-BOD
            BB=BODIES(LTEST)-(BODIES(LTEST)-BODIES(LTEST+1))*COMON
            CHANGE=CHANGE+DVBBF
            IF(CHANGE .GE. BOD+1.0) GO TO 300
            GO TO 190
      300   LTEST=LTEST+1
            BOD=BOD+1.0
      190   CONTINUE
            NVS=NVS-1
      C
            GO TO 60
      210   REWIND KFILE
      C
            CALL SECOND(TIME1)
            TIM=TIME1-TIME
            PRINT 240,TIME1,TIM
            RETURN
      C
      C
      C
      220   FORMAT(*0 THE TIME AT THE START OF ENUP IS *,F12.3)
      240   FORMAT(*0 THE TIME AT THE END OF ENUP IS *,F12.3/F12.3,
           1* SECS WERE REQUIRED FOR THIS ADDITION*)
      C
            END
```

```
      SUBROUTINE EMDOWN(NPTS,ITYPE,LFILE,MFILE)                          014270
      COMMON    OLDEM(2410),OLDTR(2410),NEWEM(2410)                      014280
      COMMON/MAIN/ KFILE,KPANEL,PO,TEMPO                                 014290
      COMMON/XEMISS/     NMOL,XID(7),SECANT,PAVE,TAVE,WMOLID(7),WK(7),   014300
     C             DV,V1,V2,NLAYER                                       014310
      COMMON /OPANL/ OV1P,OV2P,ODVP,NLIMO                                014320
      COMMON /NPANL/ V1P,V2P,DVP,NLIM                                    014330
      DIMENSION EMSAVE(4),A1(10),A2(10),A3(10),A4(10),TRSAVE(4)          014340
      DIMENSION TR(2410),NEWTR(1),BODIES(101)                            014350
      EQUIVALENCE (TR(1),NEWTR(1))                                       014360
      REAL NEWEM,NEWTR                                                   014370
C                                                                        014380
C ************************************************************           014390
C *****THIS SUBROUTINE DOES THE LAYER ADDITION FOR THE EMISSION *******   014400
C *****LOOKING FROM SPACE TO GROUND                            *******   014410
C ************************************************************           014420
C                                                                        014430
      EXPF(Y)=EXP(AMAX1(-174.0,Y))                                       014440
      BBODY(X)=(1.1910E-12*X**3)/(EXPF(1.43789*X/TAVE)-1.0)              014450
      EXPMIN=EXP(-174.0)                                                 014460
      CALL SECOND(TIME)                                                  014470
      PRINT 220,TIME                                                     014480
      IKNL=1                                                             014490
      PRINT *," THIS OUTPUT IS WRITTEN ON FILE ",MFILE                   014500
C                                                                        014510
      PRINT *, " EMDOWN, MFILE, LFILE,KFILE  ",MFILE,LFILE,KFILE         014520
      KNL=KPANEL                                                         014530
 20   REWIND MFILE                                                       014540
      REWIND LFILE                                                       014550
      REWIND KFILE                                                       014560
      BUFFER IN(LFILE,1) (XID(1),XID(2))                                 014570
      IF (UNIT(LFILE) .EQ. 0 ) STOP                                      014580
      BUFFER IN(KFILE,1)(XID(1),NLAYER)                                  014590
      IF(UNIT(KFILE) .EQ. 0) STOP                                        014600
      BUFFER OUT(MFILE,1)(XID(1),NLAYER)                                 014610
      IF(UNIT(MFILE) .EQ. 0) STOP                                        014620
      DO 30 K=1,4                                                        014630
      TRSAVE(K)=0.                                                       014640
   30 EMSAVE(K)=0.                                                       014650
      ATYPE=ITYPE                                                        014660
      IF(ITYPE .NE. 0) GO TO 430                                         014670
C                                                                        014680
C 1/1 RATIO ONLY                                                         014690
C                                                                        014700
      JUG=3                                                              014710
      JOG=2                                                              014720
      GO TO 360                                                          014730
      BUFFER IN(LFILE,1)(OV1P,NLIMO)                                     014740
      IF(UNIT(LFILE) .EQ. 0) STOP                                        014750
      BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))                          014760
      IF(UNIT(LFILE) .EQ. 0) STOP                                        014770
      BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))                          014780
      IF(UNIT(LFILE) .EQ. 0) STOP                                        014790
      DO 536 I=1,NLIM                                                    014800
      NEWEM(I)=TR(I)*OLDEM(I)+(1.0-TR(I))*BB                             014810
      NEWTR(I)=TR(I)*OLDTR(I)                                            014820
      TESTER=TESTER+DVP                                                  014830
      IF(TESTER .LT. CHANGE) GO TO 530                                   014840
      CONCN=CHANGE-BOD                                                   014850
      BB=BODIES(LTEST)-(BODIES(LTEST)-BODIES(LTEST+1))*CONCN             014860
      CHANGE=CHANGE+DVBBF                                                014870
      IF(CHANGE .GE. BOD+1.0) GO TO 560                                  014880
      GO TO 530                                                          014890
 560  LTEST=LTEST+1                                                      014900
      BOD=BOD+1.0                                                        014910
 530  CONTINUE                                                           014920
      GO TO 190                                                          014930
 550  IKNL=IKNL+1                                                        014940
      IF(IKNL .LE. KNL) GO TO 360                                        014950
      GO TO 210                                                          014960
C                                                                        014970
C ALL RATIOS EXCEPT 1/1                                                  014980
C                                                                        014990
 430  LL=ITYPE+1                                                         015000
      JOG=1                                                              015010
```

```
      AP=1.0/(ATYPE+1.0)
      DO 40 JPG=1,ITYPE
      APG=JPG
      P=1.0-(AP*APG)
      PRINT *," P= ",P
C ITYPE IS THE NUMERATOR OF THE RATIO OF OLDDV TO NEWDV WHICH IS CLOSEST
C TO A 1/1,2/1,3/1 ETC TO 20/1.  P EQUALS THE RECIPROCAL OF THIS RATIO.
C
C *******  THE FOLLOWING ARE THE CONSTANTS FOR THE LAGRANGE 4 POINT I
C       INTERPOLATION                                  ******************
      A1(JPG)=-P*(P-1.0)*(P-2.0)/6.0
      A2(JPG)=(P**2-1.0)*(P-2.0)*0.5
      A3(JPG)=-P*(P+1.0)*(P-2.0)*0.5
      A4(JPG)=P*(P**2-1.0)/6.0
   40 CONTINUE
C
C *************  BEGINNING OF LOOP THAT DOES ADDITION    ****************
C
      JUG=1
$     READ(LFILE) OV1P,OV2P,ODVP,NLIMO
      BUFFER IN(LFILE,1)(OV1P,NLIMO)
      IF(UNIT(LFILE) .EQ. 0) STOP
C
C TRANSMISSION AT THE PRESENT LAYER
C
$     READ(LFILE)(OLDEM(J),J=1,NLIMO)
  400 BUFFER IN(LFILE,1)(OLDEM(1),OLDEM(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP
      BUFFER IN(LFILE,1)(OLDTR(1),OLDTR(NLIMO))
      IF(UNIT(LFILE) .EQ. 0) STOP
  390 NVS=1
      GO TO 360
360   NEWEM(1)=TR(1)*OLDEM(1)+(1.0-TR(1))*BB
      NEWTR(1)=TR(1)*OLDTR(1)
      TESTER=TESTER+DVP
      II=2
   60 JJ=1
C
C
      DO 100 JPG=1,LL
      IF(JPG .EQ. LL) GO TO 70
      IF(NVS .EQ. 1) GO TO 80
      GO TO 90
70    NEWEM(II)=TR(II)*OLDEM(NVS)+(1.0-TR(II))*BB
      NEWTR(II)=TR(II)*OLDTR(NVS)
      TESTER=TESTER+DVP
      GO TO 110
   80 IF(EMSAVE(4) .EQ. 0.0) GO TO 81
      EMSTOR=EMSAVE(4)
      TRSTOR=TRSAVE(4)
      GO TO 100
   81 EMSTOR=OLDEM(1)
      TRSTOR=OLDTR(1)
      GO TO 100
   90 EMSTOR=OLDEM(NVS-1)
      TRSTOR=OLDTR(NVS-1)
C
C INTERPOLATION OF THE PRESENT TRANSMISSION
C
  100 OLDEMI=A1(JJ)*EMSTOR+A2(JJ)*OLDEM(NVS)+A3(JJ)*OLDEM(NVS+1)+
     1 A4(JJ)*OLDEM(NVS+2)
      OLDTRI=A1(JJ)*TRSTOR+A2(JJ)*OLDTR(NVS)+A3(JJ)*OLDTR(NVS+1)+
     1A4(JJ)*OLDTR(NVS+2)
      NEWEM(II)=(1.0-TR(II))*BB+TR(II)*OLDEMI
      NEWTR(II)=TR(II)*OLDTRI
      TESTER=TESTER+DVP
110   NVS=NVS+1
      IF(NVS .LE. NLIMO-2) GO TO 140
      EMSAVE(1)=OLDEM(NVS-1)
      EMSAVE(2)=OLDEM(NVS)
      EMSAVE(3)=OLDEM(NVS+1)
      EMSAVE(4)=OLDEM(NVS-2)
      TRSAVE(1)=OLDTR(NVS-1)
      TRSAVE(2)=OLDTR(NVS)
      TRSAVE(3)=OLDTR(NVS+1)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
      TRSAVE(4)=OLDTR(NVS-2)
$     READ(LFILE) OV1P,OV2P,ODVP,NLIMO
$     IF(EOF(LFILE)) 180,120
      BUFFER IN(LFILE,1)(OV1P,NLIMO)
      IF(UNIT(LFILE)) 120,180,180
  120 NLIMO=NLIMO+3
$     READ(LFILE)(OLDEM(J),J=4,NLIMO)
      BUFFER IN(LFILE,1)(OLDEM(4),OLDEM(NLIMO))
      IF(UNIT(LFILE)) 42,180,180
42    BUFFER IN(LFILE,1)(OLDTR(4),OLDTR(NLIMO))
      IF(UNIT(LFILE)) 43,180,180
43    OLDEM(1)=EMSAVE(1)
      OLDEM(2)=EMSAVE(2)
      OLDEM(3)=EMSAVE(3)
      OLDTR(1)=TRSAVE(1)
      OLDTR(2)=TRSAVE(2)
      OLDTR(3)=TRSAVE(3)
      NVS=2
  140 II=II+1
      JJ=JJ+1
      IF(II .GT. NLIM) GO TO 150
      GO TO 170
C
C AN EOF HAS OCCURED ON THE FILE FOR THE PREVIOUS LAYER
C
  180 II=II+1
      EM1=(EMSAVE(3)+EMSAVE(2))*0.5
      TR1=(TRSAVE(3)+TRSAVE(2))*0.5
200   NEWEM(II)=(1.0-TR(II))*BB+TR(II)*EM1
      NEWTR(II)=TR1*TR(II)
      II=II+1
      IF(II .LE. NLIM) GO TO 200
C
C WRITE OUTPUT FILE
C
$150  WRITE(MFILE) V1P,V2P,DV,NLIM
$     WRITE(MFILE)(NEWEM(J),J=1,NLIM)
$     WRITE(MFILE)(NEWTR(J),J=1,NLIM)
  150 BUFFER OUT(MFILE,1)(V1P,NLIM)
      IF(UNIT(MFILE) .EQ. 0) STOP
      BUFFER OUT(MFILE,1)(NEWEM(1),NEWEM(NLIM))
      IF(UNIT(MFILE) .EQ. 0) STOP
      BUFFER OUT(MFILE,1)(NEWTR(1),NEWTR(NLIM))
      IF(UNIT(MFILE) .EQ. 0) STOP
      IFIVE=5
      IF(NLIM .EQ. 5) IFIVE=1
      IF ( NPTS .LE. 0 ) GO TO 333
      DO 329 J=1,NPTS
      VP=V1P+FLOAT(J-1)*DVP
329   PRINT 900, J,VP,NEWEM(J),NEWTR(J)
      JEND=NLIM-NPTS+1
      IF(JEND .LT. 0) JEND=NLIM
      DO 330 J=JEND,NLIM
      VP=V1P+FLOAT(J-1)*DVP
330   PRINT 900, J,VP,NEWEM(J),NEWTR(J)
900   FORMAT(I10,F12.6,3F15.8)
333   CONTINUE
      GO TO (545,550),JOG
545   IKNL=IKNL+1
      IF(IKNL .LE. KNL) GO TO 160
      GO TO 210
  160 JOG=2
  360 BUFFER IN(KFILE,1)(V1P,NLIM)
      IF(UNIT(KFILE) .EQ. 0) STOP
      BUFFER IN(KFILE,1)(TR(1),TR(NLIM))
      IF(UNIT(KFILE) .EQ. 0) STOP
C
C COMPUTE THE BLACK BODY EVERY 1 WAVE # BEGINNING AT V1P
C
      BEGIN=V1P
      RCM=BEGIN
      I=1
  270 BODIES(I)=BBODY(BEGIN)
      I=I+1
```

```
               IF(I .GT. IB1) STOP
               BEGIN=BEGIN+1.0
               IF(BEGIN .LE. V2P+1.0) GO TO 270
               IF(I .EQ. 2) BODIES(2)=BODIES(1)
CCC              PRINT *,(BODIES(I),I=1,10)
               COMON=1.44*V1P/TAVE
               IF(COMON .LE. 4.605) GO TO 280
               DENOM=1.0
               GO TO 290
           280 DENOM=1.0-EXPF(-COMON)
           290 DVBBF=V1P*1.0E-4/ABS(3.6-(COMON/DENOM))
               BB=BODIES(1)
               TESTER=V1P
               CHANGE=V1P+DVBBF
               LTEST=1
               DO 260 J=1,NLIM
               TRJ=TR(J)
               TR(J)=1.
               IF ( TRJ .LE. 0. ) GO TO 260
               IF ( TRJ .LT. 174.0 ) GO TO 250
               TR(J)=EXPMIN
               GO TO 260
        250    TR(J)=EXP(-TRJ)
        260    CONTINUE
           370 FORMAT(1X1P10E11.3)
               GO TO (340,350,535),JUG
           350 II=1
        170    IF(TESTER .LT. CHANGE) GO TO 190
               COMON=CHANGE-BOD
               BB=BODIES(LTEST)-(BODIES(LTEST)-BODIES(LTEST+1))*COMON
               CHANGE=CHANGE+DVBBF
               IF(CHANGE .GE. BOD+1.0) GO TO 300
               GO TO 190
           300 LTEST=LTEST+1
               BOD=BOD+1.0
        190    CONTINUE
               NVS=NVS-1
               GO TO 60
           210 REWIND KFILE
C
               CALL SECOND(TIME1)
               TIM=TIME1-TIME
               PRINT 240,TIME1,TIM
               RETURN
C
C
C
        220    FORMAT(*0 THE TIME AT THE START OF ENDOWN IS *,F12.3)
        240    FORMAT(*0 THE TIME AT THE END OF ENDOWN IS *,F12.3/F12.3,
              1* SECS WERE REQUIRED FOR THIS ADDITION*)
C
               END
```

LISTING OF HIRACD SUBROUTINES

```
      SUBROUTINE MIRACO
C
C
**************************************************************************
*                                                                        *
*                                                                        *
*     CALCULATES MONOCHROMATIC ABSORPTION COEFFICIENT FOR SINGLE LAYER  *
*                                                                        *
*                                                                        *
*             USES MIRACC ALGORITHM FOR THE DOPPLER PROFILE              *
*                                                                        *
**************************************************************************
C
C
      COMMON GNU(250),S(250),ALFA0(250),EPP(250),MOL(250),
C    EFDPTH(250),MEFALF(250)
      COMMON FF(3600)
      COMMON/MAIN/ KFILE,KPANEL,P0,TEMP0
      COMMON/MFN/     NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
     C             DV,V1,V2,NLAYER
      COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
      COMMON /MXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
     1AX
      COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
      COMMON /XPANEL/ V1P,V2P,DVP,NLIM,NSHIFT,NPTS
      COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
      DIMENSION ALFCOR(7),ALFD1(7),U(7),SCOR(7)
      DIMENSION XO(251)
C
      DATA NWF / 4/, DXF / 0.02/, NF /201/, NFMAX /251/
      DATA NWS /16/, DXS / 0.08/, NS /201/, NSMAX /251/
      DATA NWVS/64/, DXVS/ 0.32/, NVS/201/, NVSMAX/251/
C
      DATA IENTER/0/, LIMIN/250/, NSHIFT/32/, NLIMF/2401/, NPTS/   0/
C
      DATA  SUBID /10H  MIRACO   /
C
      IF (IENTER.NE.0) GO TO 10
      IENTER=1
      NLIMS=(NLIMF/4)+1
      NLIMVS=(NLIMS/4)+1
C     NOTE (DXVS/DXF) IS 16 AND (DXVS/DXS) IS 4
      NBOUND=(DXVS/DXF)*FLOAT(NWVS)
      MAXF=NLIMF+NBOUND
      MAXS=(MAXF/4)+1
      MAXVS=(MAXS/4)+1
      CALL SHAPED(XO)
      CALL MOLEC(1,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
   10 PRINT 900
      PRINT 970, SUBID
      TIMRDF=TIMCNV=TIMPNL=0.
      PRINT 905,     (XID(I),I=1,7)
      REWIND KFILE
      IEOF=0
      PRINT 910, KFILE
      PRINT 915, SECANT
      PRINT 925, PAVE,TAVE
      PRINT 930, (HMOLID(M),WK(M),M=1,NMOL)
      PRINT 940, DV,V1,V2
      DVF=DV
      DVS=(DXS/DXF)*DV
      DVVS=(DXVS/DXF)*DV
      BOUND=FLOAT(NBOUND)*DV/2.
      BOUNDF=BOUND*(DXF/DXVS)/2.
      PRINT 942, BOUNDF
      ALFMAX=BOUND/FLOAT(NWF)
      NLO=NBOUND+1
      NHI=NLIMF+NSHIFT-1
      DO 50 I=1,MAXF
   50 FF(I)=0.
C
     $ WRITE (KFILE) (XID(I),I=1,7),SECANT,PAVE,TAVE,(HMOLID(M),M=1,7),
     $ *  (WK(I),I=1,NMOL),DV,V1,V2
      BUFFER OUT(KFILE,1) (XID(1),NLAYER)
```

```
      IF (UNIT(KFILE).EQ.0) STOP "99"
C
      VFT=V1-2.*BOUND
      VBOT=V1-BOUND
      VTOP=V2+BOUND
C
      XKT0=0.6951*TEMP0
      XKT =0.6951*TAVE
      XKTFAC=(1./XKT) - (1./XKT0)
      CALL MOLEC(2,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
      DO 80 M=1,NMOL
   80 U(M)=WK(M)*SCOR(M)*SECANT
      ICNT=0
      SUMALF=0.
      NCHNG=0
C
   90 CONTINUE
C
  100 CONTINUE
C
      CALL SECOND (TIME0)
      IF (IEOF.NE.0) GO TO 150
      CALL RDFILE(GNU,S,ALFA0,EPP,MOL)
      CALL SECOND (TIME)
      TIMRDF=TIMRDF+TIME-TIME0
C
      IF(IEOF.NE.0) GO TO 140
C
C     MODIFY LINE DATA FOR TEMPERATURE,PRESSURE, AND COLUMN DENSITY
C
      DO 130 I=ILO,IHI
      M=MOL(I)
      EFDPTH(I)=S(I)*U(M)
      IF (EFDPTH(I).LE.0.) GO TO 130
      ICNT=ICNT+1
      ALFD=GNU(I)*ALFD1(M)
      IF(ALFD.GE.DV) GO TO 110
      PRINT 945, GNU(I),S(I),ALFA0(I),ALFD,DV,M
      ALFD=DV
      NCHNG=NCHNG+1
  110 IF (ALFD.LE.ALFMAX) GO TO 120
      PRINT 950, GNU(I),S(I),ALFA0(I),ALFD,ALFMAX,M
      ALFD=ALFMAX
      NCHNG=NCHNG+1
  120 CONTINUE
      SUMALF=SUMALF+ALFD
      RECALF(I)=1./ALFD
      EFDPTH(I)=EFDPTH(I)*EXP(-EPP(I)*XKTFAC)*RECALF(I)*
     $          (1.-EXP(-GNU(I)/XKT))/(1.-EXP(-GNU(I)/XKT0))
  130 CONTINUE
      IF (NCHNG.GT.100) GO TO 160
  140 CONTINUE
C
      CALL CONVFN0(GNU,EFDPTH,RECALF,MOL,FF,XD)
C
      IF(IPANEL.EQ.0) GO TO 100
C
  150 CALL PANEL0 (FF,KFILE)
C
      CALL SECOND (TIME)
      KPANEL=KPANEL+1
      IF (ISTOP.NE.1) GO TO 140
      END FILE KFILE
  160 CONTINUE
      PRINT 955,TIME,TIMRDF,TIMCNV,TIMPNL
      IF (ICNT.LE.0) GO TO 170
      AVALF=SUMALF/FLOAT(ICNT)
      PRINT 960,AVALF,ICNT,NCHNG
C
  170 RETURN
C
C
C
  900 FORMAT (1H0)
  904 FORMAT(1X,7A10)
  910 FORMAT(*0 OUTPUT FILE NO. =* I5)
```

THIS PAGE IS BEST QUALITY PRACTICABLE FROM COPY FURNISHED TO DDC


```
  915  FORMAT(*0 SECANT =* F15.5)
  920  FORMAT (10(1PE10.3))
  925  FORMAT(*0 PRESS(MB) =* F12.5/ *0 TEMP(K) =* F11.2)
  930  FORMAT(23H0 COLUMN DENSITY(MOLECULES/CM**2)  ,
      1  /    15X,A6, * = * 1PE10.3) )
  940  FORMAT(*0 DV(CM-1) = *F12.8/*0 V1(CM-1) = *F12.4/*0 V2(CM-1) = *
      1  F12.4)
  942  FORMAT(17H0 BOUNDF (CM-1) = ,F8.4)
  945  FORMAT (* ---------------------------*F10.4,E14.3,3F10.6,I5)
  950  FORMAT (* ++++++++++++++++++++++++++++*F10.4,E14.3,3F10.6,I5)
  955  FORMAT (1H0,10X,*TIME*,11X,*READ*,4X,*CONVOLUTION*,10X,*PANEL*/
      1 4F15.3)
  960  FORMAT (*0  AVERAGE WIDTH = *F10.5,
      C *    NO. LINES = *I10,  *     NO. CHANGES = *I10  )
  970  FORMAT(1H0,9X,A10)
       END
```



```
       SUBROUTINE CONVFND(GNU,EFDPTH,RECALF,MOL,FF,XD)
       COMMON/NEW/      NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
      C               DV,V1,V2,NLAYER
       COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
       COMMON /MXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
      1AX
       COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLC,NPI,DVS,CVVS
       COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
       DIMENSION GNU(1), EFDPTH(1), RECALF(1), FF(1)
       DIMENSION MOL(1)
       DIMENSION XD(1)
       CALL SECOND (TIMEO)
       RATVX=CVVS/DXVS
       FVS=FLOAT(NWVS)/DXVS
       CONF=CVVS/DV-1.
       IF (ILO.GT.IHI) GO TO 50
       DO 40 I=ILO,IHI
       DEPTHI=EFDPTH(I)
       IF(DEPTHI.LE.0.) GO TO 40
       ZSLOPE=RECALF(I)*RATVX
       ZINT=(GNU(I)-VFT)/DVVS
       BFVS=FVS/ZSLOPE
       JMAX=(ZINT+BFVS)+2.
       IF (JMAX.LE.MAXVS) GO TO 20
       ILAST=I-1
       GO TO 50
  20   JMIN=(ZINT-BFVS)+2.
       JMINF=(ZINT*CONF)
       ZVS=(FLOAT(JMIN-2)-ZINT)*ZSLOPE
       ZF=ZVS+(FLOAT(JMINF)-ZINT*CONF)*ZSLOPE
       DO 30 JJ=JMIN,JMAX
       JF=JMINF+JJ
       ZF=ZF+ZSLOPE
       IZF=ABS(ZF)+1.5
       FF(JF)=FF(JF)+DEPTHI*XD(IZF)
  30   CONTINUE
  40   CONTINUE
       ILAST=IHI
  C     IDATA=0 FOR MORE DATA REQUIRED ** IDATA=1 IF NO MORE DATA REQUIRED
       IPANEL=IDATA
       GO TO 60
  50   CONTINUE
       IPANEL=1
  60   ILO=ILAST+1
       CALL SECOND (TIME)
       TIMCNV=TIMCNV+TIME-TIMEO
       RETURN
  C
       END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC


```
                SUBROUTINE PANELD (FF,KFILE)                                        002250
                COMMON/NEW/     NMOL,XID(7),SECANT,PAVE,TAVE,MMOLID(7),WK(7),       002260
               C                DV,V1,V2,NLAYER                                     002270
                COMMON /XSUB/LIMIN,ILC,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,ICATA     002280
                COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS    002290
                COMMON /XPANEL/ V1P,V2P,DVP,NLIM,NSHIFT,NPTS                         002300
                COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL                               002310
                DIMENSION FF(1)                                                     002320
                CALL SECOND (TIME0)                                                 002330
                ISTOP=0                                                             002340
                NNHI=(V2-VFT)/DV+1.5                                                002350
                IF (NHI.GE.NNHI) ISTOP=1                                            002360
                IF (ISTOP.EQ.1) NHI=NNHI                                            002370
                JMXPR=NLO+NPTS                                                      002380
                JJPR=NHI-NPTS                                                       002390
                LIMLO=(NLO-1)/4-3                                                   002400
                LIMHI=(NHI/4)+1                                                     002410
                IF (NPTS.EQ.0) GO TO 60                                             002420
                DO 40 J=NLO,JMXPR                                                   002430
                VP=VFT+FLOAT(J-1)*DV                                                002440
         40     PRINT 900, J,FF(J),VP                                               002450
                DO 50 J=JJPR,NHI                                                    002460
                VP=VFT+FLOAT(J-1)*DV                                                002470
         50     PRINT 900, J,FF(J),VP                                               002480
         60     CONTINUE                                                            002490
                NLIM=NHI-NLO+1                                                      002500
                V1P=VFT+FLOAT(NLO-1)*DV                                             002510
                V2P=VFT+FLOAT(NHI-1)*DV                                             002520
         C      V1P IS FIRST FREQ OF PANELV                                         002530
         C      V2P IS LAST FREQ OF PANELV                                          002540
         $      WRITE (KFILE) V1P,V2P,DVP,NLIM                                      002550
         $      WRITE (KFILE) (FF(J),J=NLO,NHI)                                     002560
                BUFFER OUT (KFILE,1) (V1P,NLIM)                                     002570
                IF (UNIT(KFILE) .EQ. 0 ) STOP                                       002580
                BUFFER OUT (KFILE,1) (FF(NLO),FF(NHI))                              002590
                IF (UNIT(KFILE) .EQ. 0 ) STOP                                       002600
                VFT=VFT+FLOAT(NLIMF-1)*DV                                           002610
                IF (ISTOP.EQ.1) GO TO 140                                           002620
                JF=1                                                                002630
                DO 80 J=NLIMF,MAXF                                                  002640
                FF(JF)=FF(J)                                                        002650
         80     JF=JF+1                                                             002660
                DO 90 J=JF,MAXF                                                     002670
         90     FF(J)=0.                                                            002680
                NLO=NSHIFT+1                                                        002690
         140    CALL SECOND (TIME)                                                  002700
                TIMPNL=TIMPNL+TIME-TIME0                                            002710
                RETURN                                                              002720
         C                                                                          002730
         C                                                                          002740
         C                                                                          002750
         900    FORMAT (I10,24X,E12.5,F12.5)                                        002760
                END                                                                 002770
```

LISTING OF HIRACL SUBROUTINES

```
      SUBROUTINE HIRACL
C
C
*************************************************************************
*                                                                       *
*                                                                       *
*     CALCULATES MONOCHROMATIC ABSORPTION COEFFICIENT FOR SINGLE LAYER *
*                                                                       *
*                                                                       *
*              USES HIRACC ALGORITHM FOR THE LORENTZ PROFILE            *
*                                                                       *
*************************************************************************
C
C
      COMMON GNU(250),S(250),ALFA0(250),EPP(250),MOL(250),
     C        EPDPTH(250),RECALF(250)
      COMMON FF(3600),SF(900),VSF(225)
      COMMON/MAIN/ KFILE,KPANEL,P0,TEMP0
      COMMON/NEW/      NMOL,XID(7),SECANT,PAVE,TAVE,HMOLID(7),WK(7),
     C            DV,V1,V2,NLAYER
      COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
      COMMON /MXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
     1AX
      COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
      COMMON /XPANEL/ V1P,V2P,DVP,NLIM,NSHIFT,NPTS
      COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
      DIMENSION ALFCOR(7),ALFD1(7),U(7),SCOR(7)
      DIMENSION XF(251),XS(251),XVS(251)
C
      DATA NWF / 4/, DXF / 0.02/, NF /201/, NFMAX /251/
      DATA NWS /16/, DXS / 0.08/, NS /201/, NSMAX /251/
      DATA NWVS/64/, DXVS/ 0.32/, NVS/201/, NVSMAX/251/
C
      DATA IENTER/0/, LIMIN/250/, NSHIFT/32/, NLIM/ 01/, NPTS/ 0/
C
      DATA SUBID /10H  HIRACL  /
C
      IF (IENTER.NE.0) GO TO 10
      IENTER=1
      NLIMS=(NLIMF/4)+1
      NLIMVS=(NLIMS/4)+1
C     NOTE (DXVS/DXF) IS 16 AND (DXVS/DXS) IS 4
      NBOUND=FLOAT(NWVS)*(DXVS/DXF)
      MAXF=NLIMF+NBOUND
      MAXS=(MAXF/4)+1
      MAXVS=(MAXS/4)+1
      CALL SHAPEL(XF,XS,XVS)
      CALL MOLEC(1,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
   10 PRINT 900
      PRINT 970, SUBID
      TIMRDF=TIMCNV=TIMPNL=0.
      PRINT 904,     (XID(I),I=1,7)
      REWIND KFILE
      IEOF=0
      PRINT 910, KFILE
      PRINT 915, SECANT
      PRINT 925, PAVE,TAVE
      PRINT 930, (HMOLID(M),WK(M),M=1,NMOL)
      PRINT 940, DV,V1,V2
      DVP=DV
      DVS=(DXS/DXF)*DV
      DVVS=(DXVS/DXF)*DV
      BOUND=FLOAT(NBOUND)*DV/2.
      BOUNVS=BOUND/2.
      PRINT 942, BOUNVS
      ALFMAX=BOUND/FLOAT(NWVS)
      NLO=NBOUND+1
      NHI=NLIMF+NSHIFT-1
      DO 50 I=1,MAXF
   50 FF(I)=0.
      DO 60 I=1,MAXS
   60 SF(I)=0.
      DO 70 I=1,MAXVS
   70 VSF(I)=0.
```

```
      C
      $      WRITE (KFILE) (XID(I),I=1,7),SECANT,PAVE,TAVE,(HMOLID(M),M=1,7),
      $     $  (WK(I),I=1,NMOL),DV,V1,V2
             BUFFER OUT(KFILE,1) (XID(1),NLAYER)
             IF (UNIT(KFILE).EQ.0) STOP "99"
      C
             VFT=V1-2.*BOUND
             VBOT=V1-BOUND
             VTOP=V2+BOUND
      C
             XKT0=0.6951*TEMP0
             XKT =0.6951*TAVE
             XKTFAC=(1./XKT) - (1./XKT0)
             CALL MOLEC(2,HMOLID,NMOL,TEMP0,TAVE,P0,PAVE,SCOR,ALFCOR,ALFD1)
             DO 80 M=1,NMOL
      80     U(M)=WK(M)*SCOR(M)*SECANT
             ICNT=0
             SUMALF=0.
             NCHNG=0
      C
      90     CONTINUE
      C
      100    CONTINUE
      C
             CALL SECOND (TIME0)
             IF (IEOF.NE.0) GO TO 150
             CALL RDFILE(GNU,S,ALFA0,EPP,MOL)
             CALL SECOND (TIME)
             TIMRDF=TIMRDF+TIME-TIME0
      C
             IF(IEOF.NE.0) GO TO 140
      C
      C      MODIFY LINE DATA FOR TEMPERATURE,PRESSURE, AND COLUMN DENSITY
      C
             DO 130 I=ILO,IHI
             M=MOL(I)
             EFDPTH(I)=S(I)*U(M)
             IF (EFDPTH(I).LE.0.) GO TO 130
             ICNT=ICNT+1
             ALFL=ALFA0(I)*ALFCOR(M)
             IF(ALFL.GE.DV) GO TO 110
             PRINT 945, GNU(I),S(I),ALFA0(I),ALFL,DV,M
             ALFL=DV
             NCHNG=NCHNG+1
      110    IF (ALFL.LE.ALFMAX) GO TO 120
             PRINT 950, GNU(I),S(I),ALFA0(I),ALFL,ALFMAX,M
             ALFL=ALFMAX
             NCHNG=NCHNG+1
      120    CONTINUE
             SUMALF=SUMALF+ALFL
             RECALF(I)=1./ALFL
             EFDPTH(I)=EFDPTH(I)*EXP(-EPP(I)*XKTFAC)*RECALF(I)*
      $               (1.-EXP(-GNU(I)/XKT))/(1.-EXP(-GNU(I)/XKT0))
      130    CONTINUE
             IF (NCHNG.GT.100) GO TO 160
      140    CONTINUE
      C
             CALL CONVFN(GNU,EFDPTH,RECALF,MOL,FF,SF,VSF,
            C XF,XS,XVS)
      C
             IF(IPANEL.EQ.0) GO TO 100
      C
      150    CALL PANEL (FF,SF,VSF,KFILE)
      C
             CALL SECOND (TIME)
             KPANEL=KPANEL+1
             IF (ISTOP.NE.1) GO TO 140
             END FILE KFILE
      160    CONTINUE
             PRINT 955,TIME,TIMRDF,TIMCNV,TIMPNL
             IF (ICNT.LE.0) GO TO 170
             AVALF=SUMALF/FLOAT(ICNT)
             PRINT 960,AVALF,ICNT,NCHNG
```

```
      C
  170   RETURN
      C
      C
      C
  900   FORMAT (1H0)
  904   FORMAT(1X,7A10)
  910   FORMAT(*0 OUTPUT FILE NO. =* I5)
  915   FORMAT(*0 SECANT =* F15.5)
  920   FORMAT (10(1PE10.3))
  925   FORMAT(*0 PRESS(MB) =* F12.5/ *0 TEMP(K) =* F11.2)
  930   FORMAT(33H0 COLUMN DENSITY(MOLECULES/CM**2)   ,
       1 /   /(5X,A6, * = * 1PE10.3) )
  940   FORMAT(*0 DV(CM-1) = *F12.8/*0 V1(CM-1) = *F12.4/*0 V2(CM-1) = *
       1  F12.4)
  942   FORMAT(17H0 BOUNDVS(CM-1) = ,F8.4)
  945   FORMAT (* ---------------------------*F10.4,E14.3,3F10.6,I5)
  950   FORMAT (* +++++++++++++++++++++++++++*F10.4,E14.3,3F10.6,I5)
  955   FORMAT (1H0,10X,*TIME*,11X,*READ*,6X,*CONVOLUTION*,10X,*PANEL*/
       1 4F15.3)
  960   FORMAT (*0  AVERAGE WIDTH = *F10.5,
       C *   NO. LINES = *I10,  *    NO. CHANGES = *I10  )
  970   FORMAT(1H0,9X,A10)
        END
```



```
        SUBROUTINE CONVFNL(GNU,EFDPTH,RECALF,MOL,FF,SF,VSF,XF,XS,XVS)
        COMMON/NEW/     NMOL,XID(7),SECANT,PAVE,TAVE,MMOLID(7),WK(7),
       C              DV,V1,V2,NLAYER
        COMMON /XSUB/LIMIN,ILO,IHI,VBOT,VTOP,VFT,IEOF,IPANEL,ISTOP,IDATA
        COMMON /WXX/ NWF,DXF,NF,NWS,DXS,NS,NWVS,DXVS,NVS,NFMAX,NSMAX,NVSM
       1AX
        COMMON /SUB1/ MAXF,MAXS,MAXVS,NLIMF,NLIMS,NLIMVS,NLO,NHI,DVS,DVVS
        COMMON/XTIME/TIME,TIMRDF,TIMCNV,TIMPNL
        DIMENSION GNU(1), EFDPTH(1), RECALF(1), FF(1), SF(1), VSF(1)
        DIMENSION MOL(1)
        DIMENSION XF(1), XS(1), XVS(1)
        CALL SECOND (TIMEO)
        RATVX=DVVS/DXVS
        FVS=FLOAT(NWVS)/DXVS
        CONF=DVVS/DV-1.
        CONS=DVVS/DVS-1.
        IF (ILO.GT.IHI) GO TO 50
        DO 40 I=ILO,IHI
        DEPTHI=EFDPTH(I)
        IF(DEPTHI.LE.0.) GO TO 40
        ZSLOPE=RECALF(I)*RATVX
        ZINT=(GNU(I)-VFT)/DVVS
        BFVS=FVS/ZSLOPE
        JMAX=(ZINT+BFVS)+2.
        IF (JMAX.LE.MAXVS) GO TO 20
        ILAST=I-1
        GO TO 50
   20   JMIN=(ZINT-BFVS)+2.
        JMINF=(ZINT*CONF)
        JMINS=(ZINT*CONS)
        ZVS=(FLOAT(JMIN-2)-ZINT)*ZSLOPE
        ZS=ZVS+(FLOAT(JMINS)-ZINT*CONS)*ZSLOPE
        ZF=ZVS+(FLOAT(JMINF)-ZINT*CONF)*ZSLOPE
        DO 30 JJ=JMIN,JMAX
        JS=JMINS+JJ
        JF=JMINF+JJ
        ZVS=ZVS+ZSLOPE
        ZS=ZS+ZSLOPE
        ZF=ZF+ZSLOPE
        IZVS=ABS(ZVS)+1.5
        IZS=ABS(ZS)+1.5
        IZF=ABS(ZF)+1.5
        VSF(JJ)=VSF(JJ)+DEPTHI*XVS(IZVS)
        SF(JS)=SF(JS)+DEPTHI*XS(IZS)
        FF(JF)=FF(JF)+DEPTHI*XF(IZF)
```

```
            30    CONTINUE                                                              002260
            40    CONTINUE                                                              002270
                  ILAST=IHI                                                             002280
            C      IDATA=0 FOR MORE DATA REQUIRED ** IDATA=1 IF NO MORE DATA REQUIRED   002290
      50          IPANEL=IDATA                                                          002300
                  GO TO 60                                                              002310
            50    CONTINUE                                                              002320
                  IPANEL=1                                                              002330
            60    ILO=ILAST+1                                                           002340
      55          CALL SECOND (TIME)                                                    002350
                  TIMCNV=TIMCNV+TIME-TIMEB                                              002360
                  RETURN                                                                002370
            C                                                                           002380
                  END                                                                   002390
```